MASTER EN SCIENCES MATHÉMATIQUES

Les problèmes fractionnels du type Min-Max : les algorithmes du type Dinkelbach

Toscano, Sandra

Award date:
2006

FUNDP
NAMUR

Faculté des Sciences
Département de Mathématique

Rempart de la Vierge, 8
B - 5000 Namur (Belgique)

Les Problèmes Fractionnels du Type Min-Max : les Algorithmes du Type Dinkelbach

Mémoire présenté pour l'obtention
du grade de
Licencié en Sciences Mathématiques
par

TOSCANO **Sandra**

Promoteur : STRODIOT Jean-Jacques

Année Académique 2005-2006

Ces quatre années d'études, finalisées par ce mémoire, ne pouvaient se terminer sans que je n'adresse mes sincères remerciements aux personnes suivantes :

Tout d'abord au professeur J.J. Strodiot, qui m'a très bien encadrée pour mener à bien la réalisation de ce mémoire et m'a donné énormément de son temps.

Les assistants qui m'ont aidée à résoudre bon nombre de problèmes, plus particulièrement Anne-Sophie et Sebastian pour leur patience et leur disponibilité.

Mes parents qui m'ont toujours soutenue depuis le début de mon cursus.

Aurélie toujours présente dans les moments difficiles et Jean-François mon petit rayon de soleil.

De façon plus générale je tiens à remercier toutes les personnes, amis et proches, qui ont cru en moi et m'ont encouragée tout au long de ces quatre années.

Un grand merci à tous…

Résumé.

Notre but est de trouver un algorithme pour résoudre le problème suivant

$$\bar{\lambda} = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)}{g_t(x)} \right\} \right\},$$

où X est un sous-ensemble compact non vide de $\mathbb{R}^n$, T est un espace métrique compact, $f_t(x)$, $g_t(x)$ sont des fonctions continues sur $T \times \tilde{X}$ où $\tilde{X}$ est un sous-ensemble ouvert de $\mathbb{R}^n$ contenant X, et $g_t > 0$ sur X $\forall t \in T$.

Nous utilisons l'algorithme de Dinkelbach pour le cas où T est réduit à un seul élément puis adaptons la méthode pour le cas où T possède une infinité d'éléments. Nous étudions la convergence et la vitesse de convergence des algorithmes obtenus et nous indiquons comment les implémenter.

Summary.

Our aim is to solve the problem

$$\bar{\lambda} = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)}{g_t(x)} \right\} \right\},$$

where X is a nonempty compact subset of $\mathbb{R}^n$, T is a compact metric space, $f_t(x)$, $g_t(x)$ are continuous functions on $T \times \tilde{X}$ where $\tilde{X}$ is an open subset of $\mathbb{R}^n$ containing X, and $g_t > 0$ on X for all $t \in T$.

First we use the Dinkelbach's algorithm when T has only one element and then we modify it in case of T being infinite. Convergence and rate of convergence of the obtained algorithms as well as the way they can be implemented are studied.

Table des matières

Introduction.

Il existe de nombreuses applications de programmation non linéaire où un quotient de deux fonctions doit être maximisé ou minimisé. Mais il existe aussi d'autres applications où la fonction objectif concerne plusieurs quotients de fonctions. Cette classe de problèmes est appelée programmation fractionnelle.

En 1962, Charles et Cooper ont publié un article dans lequel ils montrent qu'un programme fractionnel linéaire avec un seul quotient peut être réduit à un programme linéaire en utilisant une transformation non linéaire. L'étude de la programmation fractionnelle avec un seul quotient a largement dominé la littérature du genre jusqu'en 1980 environ. La stratégie la plus connue était d'associer à ce problème une suite de problèmes paramétriques auxiliaires ayant une structure plus simple et dont les solutions convergent vers une solution du programme fractionnel. C'est l'algorithme de Dinkelbach [2].

Dans la décennie suivante, le problème de quotients multiples, appelé problème fractionnel généralisé, a été intensivement étudié en particulier dans les travaux pionniers de Crouzeix, Ferland et Schaible [1]. Dans leurs travaux, ces auteurs ont étendu l'algorithme de Dinkelbach au cas de quotients multiples. Ils ont obtenu une théorie de convergence et ils prouvent le taux de convergence superlinéaire de leur algorithme.

Les années suivantes, de nombreux chercheurs ont essayé d'améliorer le taux de convergence des algorithmes du type Dinkelbach et d'étendre la théorie aux problèmes qui ont une infinité de quotients. En particulier, il fut observé que parfois, il est plus efficace de ne pas résoudre exactement les sous-problèmes paramétriques. Ainsi, dans les sous-problèmes, on ne considère qu'un nombre fini de quotients avec une stratégie pour mettre à jour cet ensemble. Une autre difficulté

est que ces sous-problèmes sont en général non différentiables. En effet, la fonction objectif de ces sous-problèmes est un maximum de fonctions (différentiables). La stratégie fut de "régulariser" ces sous-problèmes en utilisant la régularisation entropique. La fonction non différentiable est approximée par le dessus par une fonction différentiable avec une estimation de l'erreur.

Dans ce mémoire nous considérons le problème non linéaire suivant

$$(P) \qquad \bar{\lambda} = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)}{g_t(x)} \right\} \right\},$$

où X est un sous-ensemble compact non vide de $\mathbb{R}^n$, T est un espace métrique compact, $f_t(x), g_t(x)$ sont des fonctions continues sur $T \times \tilde{X}$ où $\tilde{X}$ est un sous-ensemble ouvert de $\mathbb{R}^n$ contenant X, et $g_t > 0$ sur X $\forall t \in T$.

Notre but est de trouver un algorithme implémentable pour résoudre le problème (P). Ce mémoire se divise en trois parties. Dans la première nous étudions les propriétés de convergence des algorithmes de Dinkelbach. Dans la section 1 nous montrons que résoudre le problème (P) revient à chercher le zéro d'une fonction F. Ensuite à la section suivante nous rappelons la procédure de Dinkelbach pour le cas $T = \{1\}$. Avec la section 3, nous étendons cette procédure au cas où T est compact et à la section 4 nous établissons la convergence superlinéaire de l'algorithme obtenu. Dans la deuxième partie nous montrons comment rendre implémentables ces algorithmes. Dans une première section nous indiquons comment résoudre en pratique un problème min-max semi-infini en utilisant une régularisation entropique, et dans une seconde section nous proposons un algorithme implémentable de type Dinkelbach pour résoudre le problème de programmation fractionnelle avec T compact. Enfin dans la dernière partie nous appliquons les algorithmes étudiés sur des exemples et détaillons les codes MATLAB utilisés pour résoudre ces exemples. Ce mémoire est basé sur les articles [3], [4], [5], [6].

Chapitre 1

Algorithme de Dinkelbach exact.

1.1 Le problème (P) et la fonction F.

Considérons le problème non linéaire suivant

$$(P) \qquad \bar{\lambda} = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)}{g_t(x)} \right\} \right\},$$

où X est un sous-ensemble compact non vide de $\mathbb{R}^n$, T est un espace métrique compact, $f_t(x), g_t(x)$ sont des fonctions continues sur $T \times \tilde{X}$ où $\tilde{X}$ est un sous-ensemble ouvert de $\mathbb{R}^n$ contenant X, et $g_t > 0$ sur X $\forall t \in T$. Notons que T peut être un ensemble fini. Ainsi, si $T = \{1, ..., m\}$, les fonctions f_t et g_t, $t \in T$ sont alors réduites aux fonctions f_i et g_i, $i = 1, ..., m$ continues sur $\tilde{X}$, et $g_i > 0$ sur X $\forall i = 1, ..., m$. Dans ce cas, le problème (P) devient

$$(P_1) \qquad \bar{\lambda} = \min_{x \in X} \left\{ \max_{1 \leq i \leq m} \left\{ \frac{f_i(x)}{g_i(x)} \right\} \right\}.$$

Pour chaque $\lambda \in \mathbb{R}$, considérons aussi le problème

$$(P_\lambda) \qquad \min_{x \in X} \left\{ \max_{t \in T} \{ f_t(x) - \lambda g_t(x) \} \right\}.$$

et notons $F(\lambda)$ sa valeur optimale. La proposition suivante montre le lien existant entre les problèmes (P) et (P_λ) et la recherche des racines de l'équation $F(\lambda) = 0$.

Proposition 1.

(a) $F(\lambda) < +\infty$ *puisque* X *est non vide.* F *est strictement décroissante et continue.*

(b) (P) *et* (P_λ) *ont toujours au moins une solution optimale.*

(c) $\bar{\lambda}$ *est fini et* $F(\bar{\lambda}) = 0$.

(d) La racine de F *est unique.*

(e) (P) *et* $(P_{\bar{\lambda}})$ *ont la même solution.*

Preuve.

(a) Puisque f_t et g_t sont continues et que T est compact, la fonction

$$(x, \lambda) \mapsto \max_{t \in T} [f_t(x) - \lambda g_t(x)]$$

est continue. Et comme X est compact, la fonction $F(\lambda) = \min_{x \in X} \max_{t \in T} [f_t(x) - \lambda g_t(x)]$ est continue sur $\mathbb{R}^n$. Montrons maintenant que F est strictement décroissante.
Soient λ_1 et λ_2 t.q. $\lambda_1 < \lambda_2$. Alors,

$$\forall t \in T \qquad \forall x \in X \qquad f_t(x) - \lambda_1 g_t(x) > f_t(x) - \lambda_2 g_t(x),$$

et

$$\begin{aligned} \forall x \in X, \qquad \max_{t \in T} [f_t(x) - \lambda_1 g_t(x)] &> \max_{t \in T} [f_t(x) - \lambda_2 g_t(x)] \\ &\geq \min_{x \in X} \max_{t \in T} [f_t(x) - \lambda_2 g_t(x)] = F(\lambda_2). \end{aligned}$$

Comme X est compact, $\exists \bar{x} \in X$ t.q. $F(\lambda_1) = \max_{t \in T} [f_t(\bar{x}) - \lambda_1 g_t(\bar{x})]$. On obtient alors que $F(\lambda_1) > F(\lambda_2)$ c'est-à-dire que F est strictement décroissante.

(b) Comme X est compact non vide, le minimum sur X des fonctions continues $\max_{t \in T} \frac{f_t(x)}{g_t(x)}$ et $\max_{t \in T} [f_t(x) - \lambda g_t(x)]$ existe et les problèmes (P) et (P_λ) ont toujours au moins une solution optimale.

(c) $\bar{\lambda} \in \mathbb{R}$ par compacité de T et X.
1) Montrons d'abord que $F(\lambda) < 0 \Leftrightarrow \lambda > \bar{\lambda}$. Supposons que $F(\lambda) < 0$. Alors,

$$\exists\, \hat{x} \in X \text{ t.q. } \max_{t \in T} [f_t(\hat{x}) - \lambda g_t(\hat{x})] < 0$$

c'est-à-dire

$$\exists\, \hat{x} \in X \text{ t.q. } \forall t \in T \; f_t(\hat{x}) - \lambda g_t(\hat{x}) < 0.$$

On a alors que

$$\lambda > \frac{f_t(\hat{x})}{g_t(\hat{x})} \; \forall t \in T.$$

D'où

$$\lambda > \max_{t\in T} \frac{f_t(\hat{x})}{g_t(\hat{x})} \geq \min_{x\in X} \max_{t\in T} \frac{f_t(x)}{g_t(x)} = \bar{\lambda}.$$

Donc $F(\lambda) < 0 \Rightarrow \lambda > \bar{\lambda}$. Inversément, si $\lambda > \bar{\lambda}$, alors $\exists \tilde{x} \in X$ t.q.

$$\max_{t\in T} \frac{f_t(\tilde{x})}{g_t(\tilde{x})} = \bar{\lambda} < \lambda.$$

Et donc $\forall t \in T$,

$$\frac{f_t(\tilde{x})}{g_t(\tilde{x})} < \lambda.$$

Ce qui implique que $f_t(\tilde{x}) - \lambda g_t(\tilde{x}) < 0 \; \forall t \in T$ et donc $\max_{t\in T} f_t(x) - \lambda g_t(x) < 0$,

$$\text{or} \quad F(\lambda) = \min_{x\in X} \max_{t\in T} f_t(x) - \lambda g_t(x) \leq \max_{t\in T} f_t(x) - \lambda g_t(x) < 0.$$

Ainsi $\lambda > \bar{\lambda} \Rightarrow F(\lambda) < 0$.

2) Soit $\bar{x}$ une solution optimale de (P). Alors $\bar{x} \in X$ et $\bar{\lambda} = \max_{t\in T} \frac{f_t(\bar{x})}{g_t(\bar{x})}$.

$$\text{D'où } \max_{t\in T} \left[f_t(\bar{x}) - \bar{\lambda} g_t(\bar{x})\right] = 0 \text{ et } F(\bar{\lambda}) = \min_{x\in X} \max_{t\in T} \left[f_t(\bar{x}) - \bar{\lambda} g_t(\bar{x})\right] \leq 0.$$

Or par 1), $F(\bar{\lambda}) < 0 \Leftrightarrow \bar{\lambda} > \bar{\lambda}$, ce qui est impossible. On en déduit que $F(\bar{\lambda}) = 0$.

(d) En effet, il est évident comme F est strictement monotone que $F(\lambda) = 0 \Rightarrow \lambda = \bar{\lambda}$. Montrons par contraposition que $\lambda \neq \bar{\lambda} \Rightarrow F(\lambda) \neq 0$. Supposons que $\lambda \neq \bar{\lambda}$, puisque F est strictement décroissante par (a), nous avons $F(\lambda) \neq F(\bar{\lambda}) = 0$ par c).

(e) 1) Montrons d'abord qu'une solution de (P) est aussi une solution de $(P_{\bar{\lambda}})$. Soit $\bar{x}$ une solution optimale du problème (P). Alors $\bar{x} \in X$ et

$$\bar{\lambda} = \max_{t\in T} \frac{f_t(\bar{x})}{g_t(\bar{x})}.$$

Alors, $\max_{t\in T}\left[f_t(\bar{x}) - \bar{\lambda} g_t(\bar{x})\right] = 0$ et $F(\bar{\lambda}) = \min_{x\in X} \max_{t\in T}\left[f_t(x) - \bar{\lambda} g_t(x)\right] \leq 0$. Or par c) on sait que $F(\bar{\lambda}) = 0$, donc $\bar{x}$ est bien solution de $(P_{\bar{\lambda}})$.

2) Montrons maintenant qu'une solution de $(P_{\bar{\lambda}})$ est une solution de (P). Supposons que $F(\bar{\lambda}) = 0$ et que $\bar{x}$ est une solution optimale de $(P_{\bar{\lambda}})$. Alors, $\max_{t\in T}\left[f_t(\bar{x}) - \bar{\lambda} g_t(\bar{x})\right] = 0$. Et donc,

$$\max_{t\in T} \frac{f_t(\bar{x})}{g_t(\bar{x})} = \bar{\lambda}.$$

Ce qui signifie que $\bar{x}$ est une solution optimale de (P).

□

Notre stratégie pour résoudre (P) sera donc de déterminer la racine de l'équation $F(\lambda) = 0$.

1.2 La procédure de Dinkelbach pour le cas $T = \{1\}$.

Quand $T = \{1\}$, la fonction $F(\lambda)$ est réduite à

$$F(\lambda) = \min_{x \in X} \{f(x) - \lambda g(x)\}$$

où X est un sous-ensemble non vide de $\mathbb{R}^n$, f et g sont des fonctions continues sur un ensemble ouvert $\tilde{X}$ de $\mathbb{R}^n$ contenant X, et $g(x) > 0 \ \forall x \in X$.

Rappel : Le sous-différentiel d'une fonction convexe $f : \mathbb{R}^n \to \mathbb{R}$ en $x \in \mathbb{R}^n$ est l'ensemble

$$\partial f(x) = \left\{x^* \in \mathbb{R}^n | f(y) \geq f(x) + x^{*T}(y - x) \ \forall y \in \mathbb{R}^n\right\}.$$

Les éléments de $\partial f(x)$ sont appelés sous-gradients de f en x.

F étant le minimum de fonctions linéaires, elle est concave en λ sur $\mathbb{R}$, nous allons voir avec la proposition suivante qu'il est facile de trouver un sous-gradient de la fonction convexe $-F$.

Proposition 2.
Soit un certain $\hat{\lambda} \in \mathbb{R}$ et soit $\hat{x}$ une solution optimale de $\min_{x \in X} \left\{f(x) - \hat{\lambda} g(x)\right\}$.
Alors $g(\hat{x}) \in \partial(-F)(\hat{\lambda})$.

Preuve.
$\forall \lambda \in \mathbb{R}$, on a

$$\begin{aligned} F(\lambda) &= \min_{x \in X} \{f(x) - \lambda g(x)\} \leq f(\hat{x}) - \lambda g(\hat{x}) = f(\hat{x}) - \lambda g(\hat{x}) + \hat{\lambda} g(\hat{x}) - \hat{\lambda} g(\hat{x}) \\ &= f(\hat{x}) - \hat{\lambda} g(\hat{x}) - (\lambda - \hat{\lambda}) g(\hat{x}) = F(\hat{\lambda}) - (\lambda - \hat{\lambda}) g(\hat{x}), \end{aligned}$$

d'où

$$-F(\lambda) \geq -F(\hat{\lambda}) + g(\hat{x})(\lambda - \hat{\lambda}).$$

Ce qui signifie que $g(\hat{x})$ appartient au sous-différentiel de $-F$ en $\hat{\lambda}$. □

A partir de ce résultat, et puisque les sous-gradients généralisent les dérivées pour les fonctions convexes, la méthode de Newton peut être utilisée afin de trouver une racine de $F(\lambda) = 0$. Considérons la suite générée de la façon suivante

$$\lambda_{k+1} = \lambda_k - \frac{F(\lambda_k)}{-g(x^k)}$$

où x^k est une solution optimale du problème $\min_{x \in X} \{f(x) - \lambda_k g(x)\}$. Par définition de x^k, l'itération de Newton peut s'écrire

$$\lambda_{k+1} = \lambda_k + \frac{f(x^k) - \lambda_k g(x^k)}{g(x^k)} = \frac{f(x^k)}{g(x^k)}.$$

Cette procédure est exactement la procédure de Dinkelbach pour résoudre le problème fractionnel lorsque $T = \{1\}$.

L'algorithme de Dinkelbach pour $T = \{1\}$ (DA)
PAS 0 : Soit $x^0 \in X$, $\lambda_1 = f(x^0)/g(x^0)$, et $k = 1$.
PAS 1 : Déterminer une solution optimale x^k de $\min_{x \in X} \{f(x) - \lambda_k g(x)\}$.
PAS 2 : Si $F(\lambda_k) = 0$, x^k est une solution optimale de (P) et λ_k est la valeur optimale. STOP.
PAS 3 : $\lambda_{k+1} = f(x^k)/g(x^k)$. Remplacer k par $k+1$ et retourner au pas 1.

Nous allons voir que la suite $\{\lambda_k\}$ générée par l'algorithme converge linéairement dans la proposition 4.

Lemme 3. *Si la suite $\{\lambda_k\}$ est décroissante alors la suite $\{g(x^k)\}$ est aussi décroissante.*

Preuve.
Soit $\lambda_{k+1} \leq \lambda_k$, montrons que $g(x^{k+1}) \leq g(x^k)$. Soit $\{x^k\}$, la suite générée par l'algorithme de Dinkelbach pour $T = 1$. Par la proposition 2, $g(x^k) \in \partial(-F)(\lambda_k)$. Notons $\theta = -F$.

$$g(x^k) \in \partial\theta(\lambda_k) \Longrightarrow \theta(\lambda) \geq \theta(\lambda_k) + g(x^k)(\lambda - \lambda_k) \qquad \forall \lambda$$

$$g(x^{k+1}) \in \partial\theta(\lambda_{k+1}) \Longrightarrow \theta(\lambda) \geq \theta(\lambda_{k+1}) + g(x^{k+1})(\lambda - \lambda_{k+1}) \qquad \forall \lambda$$

Donc en particulier pour λ_{k+1} et λ_k

$$\theta(\lambda_{k+1}) \geq \theta(\lambda_k) + g(x^k)(\lambda_{k+1} - \lambda_k) \tag{1.1}$$

$$\theta(\lambda_k) \geq \theta(\lambda_{k+1}) + g(x^{k+1})(\lambda_k - \lambda_{k+1}) \tag{1.2}$$

Sommons (1.1) et (1.2) :

$$0 \geq \underbrace{[\lambda_k - \lambda_{k+1}]}_{\geq 0} \left[g(x^{k+1}) - g(x^k)\right]$$

D'où, $g(x^{k+1}) - g(x^k) \leq 0$ et g est bien décroissant. □

Proposition 4.

Notons M l'ensemble des solutions optimales de (P),
$\bar{\lambda^\star} = \sup\{g(x) \mid x \in M\}$ *et* $N = \{x \in M \mid g(x) = \bar{\lambda^\star}\}$. *Alors M et N sont non vides et compacts. Les suites $\{\lambda_k\}, \{g(x^k)\}$ et $\{f(x^k)\}$ sont décroissantes et convergent vers $\bar{\lambda}, \bar{\lambda^\star}$ et $\bar{\lambda}\bar{\lambda^\star}$, respectivement. De plus, chaque sous-suite convergente de $\{x^k\}$ converge vers un point de N et on a la relation suivante :*

$$0 \leq \lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda})(1 - \bar{\lambda^\star}/g(x^k)).$$

Il en découle que la suite $\{\lambda_k\}$ converge superlinéairement vers $\bar{\lambda}$.

Preuve.

La fonction F, est concave. Etant de plus finie, strictement décroissante et de racine unique (par la proposition 1), on sait alors que la suite $\{\lambda_k\}$ générée par la méthode de Newton converge vers la racine $\bar{\lambda}$ de F. Notons $\theta = -F$. Alors θ est convexe et strictement croissante.
Puisque par construction $F(\lambda_k) = f(x^k) - \lambda_k g(x^k)$ et $\lambda_{k+1} = \frac{f(x^k)}{g(x^k)}$, on a donc $\forall k$

$$\theta(\lambda_k) = -F(\lambda_k) = -f(x^k) + \lambda_k g(x^k) = (\lambda_k - \lambda_{k+1})g(x^k). \tag{1.3}$$

Comme $g(x^k) \in \partial\theta(\lambda_k)$ (proposition 2), nous avons

$$0 = \theta(\bar{\lambda}) \geq \theta(\lambda_k) + (\bar{\lambda} - \lambda_k)g(x^k). \tag{1.4}$$

En remplacant la valeur de $\theta(\lambda_k)$ de (1.3) dans (1.4), nous obtenons que $\forall k\ 0 \geq (\bar{\lambda} - \lambda_{k+1})g(x^k)$. Or $g(x^k) > 0 \Rightarrow \bar{\lambda} - \lambda_{k+1} \leq 0$ c'est-à-dire $\lambda_{k+1} \geq \bar{\lambda}\ \forall k$. Comme $\theta(\bar{\lambda}) = 0$ et que θ est croissante, on a que $\theta(\lambda_{k+1}) \geq 0$. Et par l'équation (1.3) on voit alors que $\lambda_k - \lambda_{k+1} \geq 0$ c'est-à-dire que la

suite $\{\lambda_k\}$ est décroissante.

Maintenant notons

$$\theta'_+(\bar{\lambda}) = \lim_{\lambda \to \bar{\lambda}_+} \frac{\theta(\lambda) - \theta(\bar{\lambda})}{\lambda - \bar{\lambda}} > 0$$

la dérivée à droite de θ en $\bar{\lambda}$. Alors $\theta'_+(\bar{\lambda}) \in \partial\theta(\bar{\lambda})$, et

$$\theta(\lambda_k) \geq \underbrace{\theta(\bar{\lambda})}_{=0} + (\lambda_k - \bar{\lambda})\theta'_+(\bar{\lambda}) = (\lambda_k - \bar{\lambda})\theta'_+(\bar{\lambda}). \tag{1.5}$$

En utilisant (1.3) et (1.5), on a que $(\lambda_k - \lambda_{k+1})g(x^k) \geq (\lambda_k - \bar{\lambda})\theta'_+(\bar{\lambda})$,

c'est-à-dire

$$\lambda_{k+1} - \lambda_k \leq \frac{-(\lambda_k - \bar{\lambda})\theta'_+(\bar{\lambda})}{g(x^k)}.$$

$$\lambda_{k+1} - \bar{\lambda} = \lambda_{k+1} - \lambda_k + \lambda_k - \bar{\lambda} \leq \frac{-(\lambda_k - \bar{\lambda})\theta'_+(\bar{\lambda})}{g(x^k)} + \lambda_k - \bar{\lambda} = (\lambda_k - \bar{\lambda})\left(1 - \frac{\theta'_+(\bar{\lambda})}{g(x^k)}\right) \tag{1.6}$$

Par (1.4) et par le lemme 3 on a que $\frac{\theta(\bar{\lambda}) - \theta(\lambda_k)}{\bar{\lambda} - \lambda_k} \leq g(x^k) \leq g(x^{k-1}) \leq \cdots \leq g(x^1)$. Et en prenant la limite sur $\lambda_k \to \bar{\lambda}_+$, on obtient que

$$\theta'_+(\bar{\lambda}) \leq g(x^k)\ \forall k \text{c'est-à-dire } 0 < \frac{\theta'_+(\bar{\lambda})}{g(x^k)} \leq 1 \forall k.$$

On a alors par l'équation (1.6) que la suite $\{\lambda_k\}$ converge au moins linéairement vers $\bar{\lambda}$.

De plus, puisque le sous-gradient $\partial\theta$ est semi-continu supérieurement, que

$$\lambda_k \to \bar{\lambda} \quad \text{et} \quad g(x^k) \to \bar{g} \geq \theta'_+(\bar{\lambda}),$$

nous avons que $\bar{g} \in \partial\theta(\bar{\lambda})$ et donc que $\bar{g} = \theta'_+(\bar{\lambda})$. Alors

$$\left\{g(x^k)\right\} \to \theta'_+(\bar{\lambda})$$

et la convergence de la suite $\{\lambda_k\}$ est superlinéaire par (1.6). Et, puisque $f(x^k) = \lambda_{k+1}g(x^k)$, la suite $\left\{f(x^k)\right\}$ décroisse vers $\bar{\lambda}\theta'_+(\bar{\lambda}) = \bar{\lambda}\bar{\lambda}^\star$. Soit $\bar{x}$ un point limite d'une sous-suite convergente de $\left\{x^k\right\}$. Alors $f(\bar{x}) = \bar{\lambda}g(\bar{x})$, et donc $\bar{x} \in M$. De plus, $g(x^k) \to g(\bar{x})$. Mais $\left\{g(x^k)\right\}$ converge aussi vers $\theta'_+(\bar{\lambda})$, ainsi $g(\bar{x}) = \theta'_+(\bar{\lambda}) = \bar{\lambda}^\star$ et $\bar{x} \in N$. Ainsi on a bien que $\left\{g(x^k)\right\} \to \bar{\lambda}^*$. Et en remplaçant $\theta'_+(\bar{\lambda})$ par $\bar{\lambda}^*$ dans l'équation (1.6), on obtient la relation désirée. □

1.3 Extension à plusieurs quotients.

On peut modifier la procédure de Dinkelbach en choisissant un λ_{k+1} de la façon suivante :

$$\lambda_{k+1} = \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\}.$$

L'algorithme devient alors :

L'algorithme de Dinkelbach pour T compact (DTA)

PAS 0 : Soit $x^0 \in X$, $\lambda_1 = \max_{t \in T} \left\{ f_t(x^0)/g_t(x^0) \right\}$, et $k = 1$.

PAS 1 : Déterminer une solution optimale x^k de

$$\inf_{x \in X} \left\{ \max_{t \in T} \{ f_t(x) - \lambda_k g_t(x) \} \right\} \qquad (P_{\lambda_k}).$$

PAS 2 : Si $F(\lambda_k) = 0$, x^k est une solution optimale de (P) et λ_k est la valeur optimale. STOP.

PAS 3 : $\lambda_{k+1} = \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\}$. Remplacer k par $k+1$ et retourner au pas 1.

Interprétation géométrique de l'algorithme DTA.

Soit $\lambda_k > \bar{\lambda}$ et soit x^k le minimum sur X de la fonction $F(\lambda_k, x)$ définie par

$$F(\lambda, x) = \max_{t \in T} \{ f_t(x) - \lambda g_t(x) \} \quad \forall \lambda \in \mathbb{R}, \quad \text{et } x \in X.$$

Si $F(\lambda_k) < 0$, alors $F(\lambda_k, x^k) < 0$ et $F(\lambda_k) \leq F(\lambda, x^k)$ $\forall \lambda$. A l'itération k, la fonction non convexe F est approximée supérieurement par la fonction convexe $\lambda \rightarrow F(\lambda, x^k)$ et le problème complexe de trouver une racine de F est remplacé par le problème de résoudre l'équation $F(\lambda, x^k) = 0$, qui mène au pas 3 :

$$\lambda_{k+1} = \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\}.$$

Proposition 5.

(a) $\forall k \geq 1,\ \lambda_k \geq \bar{\lambda}$ *et* $F(\lambda_k) \leq 0$.

(b) Si $F(\lambda_k) = 0$, *alors* $\lambda_k = \bar{\lambda}$ *et* x^k *est une solution optimale de* (P).

(c) La suite $\{\lambda_k\}$ *est strictement décroissante.*

Preuve.

(a) Puisque $x^k \in X$, il est évident que

$$\lambda_{k+1} = \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\} \geq \min_{x \in X} \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\} = \bar{\lambda}.$$

Alors, par la proposition 1 (a), $F(\lambda_{k+1}) \leq 0$.

(b) Supposons par l'absurde, que $\lambda_k \neq \bar{\lambda}$ et comme est F est strictement décroissante (proposition 1 a)), on a que $F(\lambda_k) < 0$, ce qui est en contradiction avec l'hypothèse $F(\lambda_k) = 0$. De plus, x^k est une solution optimale de (P) par la proposition 1(e).

(c) T étant compact, notons p l'indice qui spécifie λ_{k+1}, c'est-à-dire ;

$$\lambda_{k+1} = \max_{t \in T} \left\{ \frac{f_t(x^k)}{g_t(x^k)} \right\} = \frac{f_p(x^k)}{g_p(x^k)}.$$

Alors,

$$F(\lambda_k) = \max_{t \in T} \left\{ f_t(x^k) - \lambda_k g_t(x^k) \right\} \geq f_p(x^k) - \lambda_k g_p(x^k) = (\lambda_{k+1} - \lambda_k) g_p(x^k).$$

Donc, $\lambda_{k+1} - \lambda_k \leq \frac{F(\lambda_k)}{g_p(x^k)} < 0$ puisque $F(\lambda_k) < 0$ et $g_p(x^k) > 0$. □

L'analyse de convergence de l'algorithme dans le cas de plusieurs quotients n'est pas aussi facile que dans le cas où $T = \{1\}$. De plus, le taux de convergence est plus lent. Cependant, avant de considérer la convergence de l'algorithme, nous avons besoin de démontrer plusieurs inégalités.

Pour $\lambda \in \mathbb{R}$, soit $M(\lambda)$ l'ensemble des solutions optimales de (P_λ). Posons aussi

$$\underline{g}(x) = \min_{t \in T} g_t(x), \quad \bar{g}(x) = \max_{t \in T} g_t(x), \quad x \in X.$$

Lemme 6.

Soit λ tel que $F(\lambda)$ soit fini et que $M(\lambda)$ soit non vide. Soit $x \in M(\lambda)$. Alors,

(a) $F(\mu) \leq F(\lambda) + (\lambda - \mu)\underline{g}(x)$, *si* $\mu > \lambda$,

(b) $F(\mu) \leq F(\lambda) + (\lambda - \mu)\bar{g}(x)$, *si* $\mu < \lambda$.

Preuve.

Puisque $x \in M$, nous avons $\forall t \in T$, $F(\lambda) = \max_{t\in T} \{f_t(x) - \lambda g_t(x)\} \geq f_t(x) - \lambda g_t(x)$.

Donc, $\forall \mu \in \mathbb{R}$ et $t \in T$,

$$F(\lambda) \geq -\mu g_t(x) + f_t(x) - (\lambda - \mu)g_t(x),$$

$$\Rightarrow F(\lambda) + (\lambda - \mu)g_t(x) \geq f_t(x) - \mu g_t(x). \tag{1.7}$$

Si $\mu > \lambda$, alors par l'équation (1.7)$\forall t \in T$,

$$F(\lambda) + \underbrace{(\lambda - \mu)}_{<0}\underline{g}(x) \geq F(\lambda) + (\lambda - \mu)g_t(x) \geq f_t(x) - \mu g_t(x),$$

$$\Longrightarrow F(\lambda) + (\lambda - \mu)\underline{g}(x) \geq \max_{t\in T}[f_t(x) - \mu g_t(x)] = F(\mu).$$

Si $\mu < \lambda$, alors par l'équation (1.7)$\forall t \in T$,

$$F(\lambda) + \underbrace{(\lambda - \mu)}_{>0}\bar{g}(x) \geq F(\lambda) + (\lambda - \mu)g_t(x) \geq f_t(x) - \mu g_t(x),$$

$$\Longrightarrow F(\lambda) + (\lambda - \mu)\bar{g}(x) \geq \max_{t\in T}[f_t(x) - \mu g_t(x)] = F(\mu).$$

□

Pour $\lambda \in \mathbb{R}$, $x \in M(\lambda)$, soient

$$U(x, \lambda) = \{u \in T \mid f_u(x) - \lambda g_u(x) = F(\lambda)\}.$$

$$V(x) = \left\{v \in T \mid f_v(x)/g_v(x) = \max_{t\in T} f_t(x)/g_t(x)\right\}, \ x \in X.$$

Lemme 7.

$\forall v \in V(x^k)$ et $\forall u \in U(x^k, \lambda_k)$, nous avons :

$$\begin{array}{lllll} -F(\lambda_k)/\bar{g}(x^k) & \overset{(a)}{\leq} & -F(\lambda_k)/g_v(x^k) & \overset{(b)}{\leq} & \lambda_k - \lambda_{k+1}, \\ \lambda_k - \lambda_{k+1} & \overset{(c)}{\leq} & -F(\lambda_k)/g_u(x^k) & \overset{(d)}{\leq} & -F(\lambda_k)/\underline{g}(x^k). \end{array}$$

Preuve.

Soit $v \in V(x^k)$, alors par construction de l'algorithme,

$$\lambda_{k+1} = \max_{t \in T} \frac{f_t(x^k)}{g_t(x^k)} = \frac{f_v(x^k)}{g_v(x^k)}.$$

Puisque $x^k \in M(\lambda_k)$, nous avons

$F(\lambda_k) = \max_{t \in T} \left[f_t(x^k) - \lambda_k g_t(x^k) \right] \geq f_v(x^k) - \lambda_k g_v(x^k) = g_v(x^k)(\lambda_{k+1} - \lambda_k).$

$$\Rightarrow -F(\lambda_k) \leq g_v(x^k)(\lambda_k - \lambda_{k+1}) \Rightarrow (b).$$

Soit $u \in U(x^k, \lambda_k)$, alors

$F(\lambda_k) = f_u(x^k) - \lambda_k g_u(x^k) = g_u(x^k) \left(\frac{f_u(x^k)}{g_u(x^k)} - \lambda_k \right) \leq g_u(x^k)(\lambda_{k+1} - \lambda_k).$

$$\Rightarrow -F(\lambda_k) \geq g_u(x^k)(\lambda_k - \lambda_{k+1}) \Rightarrow (c).$$

Comme $F(\lambda_k) \leq 0$ par la proposition 5(a), le (a) et (d) découlent des définitions de $\underline{g}(x)$ et $\bar{g}(x)$.

□

Proposition 8.

Si (P) a une solution optimale $\bar{x}$ et $M(\lambda_k) \neq 0, k = 1, 2, ...,$ alors

$$(\lambda_{k+1} - \bar{\lambda}) \leq (1 - \underline{g}(\bar{x})/\bar{g}(x^k))(\lambda_k - \bar{\lambda}), k = 1, 2, ...$$

Preuve.

Par le lemme 7, nous avons

$$-F(\lambda_k)/\bar{g}(x^k) \leq \lambda_k - \lambda_{k+1}$$

$$\Longleftrightarrow$$

$$\lambda_{k+1} \leq \lambda_k + F(\lambda_k)/\bar{g}(x^k)$$

$$\Longleftrightarrow$$

$$\lambda_{k+1} - \bar{\lambda} \leq \lambda_k - \bar{\lambda} + F(\lambda_k)/\bar{g}(x^k). \tag{1.8}$$

L'existence d'une solution optimale $\bar{x}$ de (P) implique que $F(\bar{\lambda}) = 0$ et $\bar{x} \in M(\bar{\lambda})$, et donc $M(\bar{\lambda}) \neq 0$. Alors on peut appliquer le lemme 6(a) avec $\mu = \lambda_k$ et $\lambda = \bar{\lambda}$, ce qui donne

$$F(\lambda_k) \leq F(\bar{\lambda}) + (\bar{\lambda} - \lambda_k)\underline{g}(\bar{x}) = (\bar{\lambda} - \lambda_k)\underline{g}(\bar{x}).$$

En réintégrant ceci dans l'équation (1.8) on obtient

$$\lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda}) + (\bar{\lambda} - \lambda_k)\underline{g}(\bar{x})/\bar{g}(x^k) = (\lambda_k - \bar{\lambda})(1 - \underline{g}(\bar{x})/\bar{g}(x^k)).$$

□

Proposition 9.
Si (P) a une solution optimale $\bar{x}$, si $M(\lambda_k) \neq 0, k = 1, 2, ...,$ et si $\sup_k \bar{g}(x^k) < +\infty$, alors $\{\lambda_k\}$ converge linéairement vers $\bar{\lambda}$.

Preuve.
Soit $G = \sup_k \bar{g}(x^k)$. Alors, par la proposition 8

$$\lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda})(1 - \underline{g}(\bar{x})/G). \tag{1.9}$$

La suite $\{\lambda_k\}$, étant décroissante et bornée inférieurement par $\bar{\lambda}$, converge vers un certain $\tilde{\lambda} \geq \bar{\lambda}$. Alors $\tilde{\lambda} = \bar{\lambda}$ car sinon, lorsque l'on prend la limite en (1.9), on a

$$\tilde{\lambda} - \bar{\lambda} \leq (\tilde{\lambda} - \bar{\lambda})(1 - \underline{g}(\bar{x})/G)$$

$$\Longleftrightarrow$$

$$1 \leq 1 - \underline{g}(\bar{x})/G$$

$$\Longleftrightarrow$$

$$0 \geq \underline{g}(\bar{x})/G,$$

Ce qui est impossible étant donné que g est une fonction srictement positive. Donc la suite $\{\lambda_k\}$ converge vers $\bar{\lambda}$ et la convergence est linéaire. □

Proposition 10.
La suite $\{\lambda_k\}$ générée par l'algorithme DTA, si elle n'est pas finie, converge linéairement vers $\bar{\lambda}$, et chaque sous-suite convergente de $\{x^k\}$ converge vers une solution optimale de (P).

Preuve.
Par la proposition 1(b) on a que (P) a une solution optimale, $M(\lambda_k) \neq 0, k = 1, 2, ...$ et $\sup_k \bar{g}(x^k) < \infty$. Alors, par la proposition 9, la suite $\{\lambda_k\}$ converge linéairement vers $\bar{\lambda}$. Maintenant, X étant compact, il existe $\{x^{k_i}\}$ une sous-suite de $\{x^k\}$ convergent vers $\tilde{x} \in X$. Et, comme par construction, $F(\lambda_{k_i}) = \max_{t \in T} \left[f_t(x^{k_i}) - \lambda_{k_i} g_t(x^{k_i})\right]$, il s'ensuit par continuité que

$$F(\bar{\lambda}) = \max_{t \in T} \left[f_t(\tilde{x}) - \bar{\lambda} g_t(\tilde{x})\right],$$

c'est-à-dire que $\tilde{x}$ est une solution du problème $(P_{\bar{\lambda}})$. Puisque $F(\bar{\lambda}) = 0$, par la proposition 1 nous avons que $\tilde{x}$ est une solution optimale de (P) par la proposition 5 b). □

Pour $T = \{1\}$, on a vu dans la proposition 4 que la suite $\{\lambda_k\}$ converge superlinéairement vers $\bar{\lambda}$. Malheureusement ce n'est plus le cas lorsque T a plus d'un élément. De plus, la proposition 8 nous montre que lorsque $T = \{1, ..., m\}$, l'algorithme devient de plus en plus lent lorsque le nombre m d'éléments dans T devient de plus en plus grand, puisque $\bar{g}$ augmente avec m. Plus il y aura de quotients concernés, plus la méthode sera lente. Nous avons vu précédemment que pour $T = \{1\}$, l'algorithme de Dinkelbach coïncide avec la méthode de Newton. Pour un T qui a plus d'un élément, la méthode de Newton peut être assez différente de celle de Dinkelbach. En général l'algorithme de Dinkelbach DTA sera plus lent que la méthode de Newton, cependant il a deux avantages :

a) Le sous-gradient de F n'a pas besoin d'être calculé.

b) La méthode de Dinkelbach converge même pour les fonctions F non concaves.

Dans la section suivante, nous allons modifier l'algorithme de Dinkelbach de façon à ce que la convergence devienne superlinéaire.

1.4 Convergence superlinéaire.

Dans le but que les inégalités du lemme 6 soient vérifiées en $\bar{\lambda}$, nous supposons que le problème (P) a une solution optimale $\bar{x}$, et nous écrivons le problème (P) sous la forme suivante

$$\bar{\lambda} = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)/g_t(\bar{x})}{g_t(x)/g_t(\bar{x})} \right\} \right\}.$$

Notons $(\bar{P}_\lambda)$ le programme correspondant à la formulation de (P) :

$$(\bar{P}_\lambda) \qquad \bar{F}(\lambda) = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x) - \lambda g_t(x)}{g_t(\bar{x})} \right\} \right\}. \tag{1.10}$$

Par le lemme 6 appliqué à $\bar{F}$ à la place de F, on obtient

$$\begin{aligned} \bar{F}(\lambda) &\leq \bar{F}(\bar{\lambda}) - \underline{\rho}(\bar{x})(\lambda - \bar{\lambda}) \quad \text{si } \lambda > \bar{\lambda} \\ \bar{F}(\lambda) &\leq \bar{F}(\bar{\lambda}) - \bar{\rho}(\bar{x})(\lambda - \bar{\lambda}) \quad \text{si } \lambda < \bar{\lambda}, \end{aligned}$$

où $\underline{\rho}(\bar{x}) = \min_{t \in T} \{g_t(\bar{x})/g_t(\bar{x})\}$ et $\bar{\rho}(\bar{x}) = \max_{t \in T} \{g_t(\bar{x})/g_t(\bar{x})\}$. Comme $\underline{\rho} = \bar{\rho} = 1$, nous retrouvons l'inégalité du sous-gradient $1 \in \partial(-F)(\bar{\lambda})$, et par analogie avec le cas $T = \{1\}$, la méthode de type Dinkelbach se réduit à la méthode de Newton dans le voisinage de $\bar{\lambda}$ et la convergence sera au moins superlinéaire.

Mais en général $\bar{x}$ n'est pas connu à priori. La stratégie à l'itération k (c'est-à-dire quand x^k doit être déterminé) est d'approximer $g_t(\bar{x})$ par $g_t(x^{k-1})$ dans (1.10). L'agorithme résultant est appelé Algorithme de Type Dinkelbach Modifié.

L'algorithme modifié de Dinkelbach pour T compact (MDTA)

PAS 0 : Soit $x^0 \in X$, $\lambda_1 = \max_{t \in T} \{f_t(x^0)/g_t(x^0)\}$, et $k = 1$.

PAS 1 : Déterminer une solution optimale x^k de

$$F_k(\lambda_k) = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x) - \lambda_k g_t(x)}{g_t(x^{k-1})} \right\} \right\} \qquad (Q_k).$$

PAS 2 : Si $F(\lambda_k) = 0$, x^k est une solution optimale de (P) et λ_k est la valeur optimale. STOP.

PAS 3 : $\lambda_{k+1} = \max_{t \in T} \{f_t(x^k)/g_t(x^k)\}$. Remplacer k par $k+1$ et retourner au pas 1.

La version modifiée a un meilleur taux de convergence. D'abord nous prouverons que la convergence est au moins linéaire, et ensuite qu'elle est superlinéaire.

Proposition 11.
(a) Si $F_k(\lambda_k) = 0$, alors $\lambda_k = \bar{\lambda}$ et x^k est une solution optimale de (P).
(b) La suite $\{\lambda_k\}$, si elle n'est pas finie, converge au moins linéairement vers $\bar{\lambda}$, et chaque sous-suite convergente de $\{x^k\}$ converge vers une solution optimale de (P).

Preuve.
(a) Le problème (P) peut être réécrit sous la forme

$$(P) \qquad \bar{\lambda} = \min_{x \in X} \max_{t \in T} \left\{ \frac{f_t(x)/g_t(x^{k-1})}{g_t(x)/g_t(x^{k-1})} \right\}, \qquad \forall k = 1, 2, ...$$

Par construction, $\lambda_k \geq \bar{\lambda}$. Alors, si $F_k(\lambda_k) = 0$, on a par la propositon 5(b) que $\lambda_k = \bar{\lambda}$ et que x^k est une solution de (P) car par construction c'est une solution de (Q_k).

(b) Soit $\bar{x}$ une solution de (P). Alors par le lemme 6 a),

$$F_k(\lambda_k) \leq F_k(\bar{\lambda}) + (\bar{\lambda} - \lambda_k) \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1}) \right] = (\bar{\lambda} - \lambda_k) \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1}) \right]. \tag{1.11}$$

Comme x^k est une solution optimale de (Q_k),

$$\begin{aligned} F_k(\lambda_k) &= \max_{t \in T} \left\{ \frac{f_t(x^k) - \lambda_k g_t(x^k)}{g_t(x^{k-1})} \right\} \\ &= \max_{t \in T} \left\{ \frac{g_t(x^k)}{g_t(x^{k-1})} \left(\frac{f_t(x^k)}{g_t(x^k)} - \lambda_k \right) \right\} \\ &\geq \frac{g_v(x^k)}{g_v(x^{k-1})} (\lambda_{k+1} - \lambda_k), \end{aligned} \tag{1.12}$$

$\forall v \in V(x^k) = \{ v \in T \mid f_v(x^k)/g_v(x^k) = \lambda_{k+1} \}$. Par (1.11) et (1.12) on obtient alors

$$\left[g_v(x^k)/g_v(x^{k-1}) \right] (\lambda_{k+1} - \lambda_k) \leq (\bar{\lambda} - \lambda_k) \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1}) \right]$$

$$\Longleftrightarrow$$

$$\lambda_{k+1} - \lambda_k \leq (\bar{\lambda} - \lambda_k) \left[g_v(x^{k-1})/g_v(x^k) \right] \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1}) \right]$$

$$\Longleftrightarrow$$

$$\lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda}) + (\bar{\lambda} - \lambda_k) \left[g_v(x^{k-1})/g_v(x^k) \right] \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1}) \right]$$

$$\Longleftrightarrow$$

$$\lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda}) \left(1 - \left[g_v(x^{k-1})/g_v(x^k)\right] \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1})\right]\right), \qquad \forall v \in V(x^k).$$

Soit $\alpha = \min_{x,y \in X} \left(\min_{l \in T} [g_l(x)/g_l(y)] \min_{t \in T} [g_t(\bar{x})/g_t(x)]\right)$. Alors $\alpha > 0$, car X est compact et les fonctions g_i sont strictement positives et continues sur X. Et donc,

$$0 \leq \lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda})(1 - \alpha). \tag{1.13}$$

Ainsi, la convergence linéaire de la suite $\{\lambda_k\}$ est démontrée. Soit $\tilde{x}$ la limite d'une sous-suite convergente de la suite $\{x^k\}$. Alors $\tilde{x} \in X$ et $\exists \hat{x} \in X$ et une suite strictement croissante d'entiers positifs $\{n_k\}$ tels que la suite (x_{n_k-1}, x_{n_k}) converge vers $(\hat{x}, \tilde{x})$. Pour $y \in X$ et $\mu \in \mathbb{R}$, définissons

$$r(y, \mu) = \min_{x \in X} \max_{t \in T} \left[(1/g_t(y))(f_t(x) - \mu g_t(x))\right].$$

De façon évidente r est continue sur $X \times \mathbb{R}$ et nous avons

$$r(x_{n_k-1}, \lambda_{n_k}) = \max_{t \in T} \left[(1/g_t(x_{n_k-1}))(f_t(x_{n_k}) - \lambda_{n_k} g_t(x_{n_k}))\right].$$

En prenant la limite en $k \to +\infty$, on obtient

$$r(\hat{x}, \bar{\lambda}) = \max_{t \in T} \left[(1/g_t(\hat{x}))(f_t(\tilde{x}) - \bar{\lambda} g_t(\tilde{x}))\right].$$

Observons que $r(\hat{x}, \lambda)$ est la fonction $F(\lambda)$ associée au problème (P) écrit sous la forme

$$\bar{\lambda} = \min_{x \in X} \max_{t \in T} \left\{ \frac{f_t(x)/g_t(\hat{x})}{g_t(x)/g_t(\hat{x})} \right\}.$$

Alors par la proposition 1 c), il suit que $r(\hat{x}, \bar{\lambda}) = 0$ et $\tilde{x}$ est une solution optimale de (P). □

Si (P) a une solution unique, alors la suite $\{x^k\}$ générée par l'algorithme est convergente. Dans ce cas la vitesse de convergence de $\{\lambda_k\}$ vers λ^* est au moins superlinéaire comme le montre la proposition suivante.

Proposition 12.

Soit la suite $\{x^k\}$ *qui converge vers* $\bar{x}$. *Alors,* $\{\lambda_k\}$ *converge superlinéairement vers* $\bar{\lambda}$. *Si, de plus, les fonctions* $g_t, t \in T$ *satisfont la condition de Lipschitz sur* X, *alors il existe une constante positive* M *telle que*

$$0 \leq \lambda_{k+1} - \bar{\lambda} \leq M(\lambda_k - \bar{\lambda}) \left[\left\| x^k - x^{k-1} \right\| + \left\| x^{k-1} - \bar{x} \right\| \right],$$

pour k *assez grand.*

Preuve.

Par la proposition 11, $\bar{x}$ est une solution optimale de (P). De l'inégalité (1.13), on a que

$$\lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda})(1 - \alpha_k), \tag{1.14}$$

$$\text{où} \quad \alpha_k = \min_{l \in T} \left[g_l(x^{k-1})/g_l(x^k) \right] \min_{t \in T} \left[g_t(\bar{x})/g_t(x^{k-1}) \right].$$

De façon évidente, $\alpha_k \to 1$ lorsque $k \to +\infty$, impliquant la convergence superlinéaire de la suite $\{\lambda_k\}$ vers $\bar{\lambda}$. Si les fonctions g_t satisfont la condition de Lipschitz sur X, il existe une constante positive L telle que

$$|g_t(x) - g_t(y)| \leq L \|x - y\|, \quad \forall x, y \in X \text{ et } t \in T.$$

$$\Longleftrightarrow$$

$$\begin{array}{ccccc} -L\|x-y\| & \leq & g_t(x) - g_t(y) & \leq & L\|x-y\| \\ \frac{-L\|x-y\|}{g_t(y)} & \leq & \frac{g_t(x)}{g_t(y)} - 1 & \leq & \frac{L\|x-y\|}{g_t(y)} \\ 1 - \frac{L\|x-y\|}{g_t(y)} & \leq & \frac{g_t(x)}{g_t(y)} & \leq & 1 + \frac{L\|x-y\|}{g_t(y)}. \end{array}$$

Puisque X et T sont compacts, il existe un nombre réel positif β t.q.

$$0 < \beta \leq g_t(x), \quad \forall x \in X \text{ et } t \in T.$$

Et donc, $\min_{t \in T} [g_t(x)/g_t(y)] \geq \max [0, 1 - (L/\beta) \|x - y\|] \ \forall x, y \in X$. Et en utilisant ceci dans l'expression de α_k, on obtient

$$\begin{array}{rcl} \alpha_k & \geq & \max \left[0, 1 - (L/\beta) \left\| x^{k-1} - x^k \right\| \right] \max \left[0, 1 - (L/\beta) \left\| \bar{x} - x^{k-1} \right\| \right] \\ & = & \max(0, 1 - (L/\beta) \left[\left\| x^{k-1} - x^k \right\| + \left\| \bar{x} - x^{k-1} \right\| \right] + \frac{L^2}{\beta^2} \left\| x^{k-1} - x^k \right\| \left\| \bar{x} - x^{k-1} \right\|) \\ & \geq & \max(0, 1 - (L/\beta) \left[\left\| x^{k-1} - x^k \right\| + \left\| \bar{x} - x^{k-1} \right\| \right]). \end{array}$$

Alors pour un k assez grand,

$$0 \leq 1 - \alpha_k \leq (L/\beta) \left[\left\| x^{k-1} - x^k \right\| + \left\| \bar{x} - x^{k-1} \right\| \right],$$

et donc par l'équation 1.14

$$\lambda_{k+1} - \bar{\lambda} \leq (\lambda_k - \bar{\lambda}) \frac{L}{\beta} \left[\left\| x^{k-1} - x^k \right\| + \left\| \bar{x} - x^{k-1} \right\| \right].$$

En prenant $M = L/\beta$, on obtient alors le résultat souhaité. □

Pour $T = \{1\}$, l'algorithme modifié du type Dinkelbach (MDTA) coïncide avec l'algorithme du type Dinkelbach (DTA) et ils convergent tous deux superlinéairement. Par contre pour T qui a plus d'un élément, l'algorithme de type dinkelbach (DTA) converge seulement linéairement alors que l'algorithme de type Dinkelbach modifié (MDTA) converge superlinéairement sans effort supplémentaire.

Chapitre 2

Algorithme de Dinkelbach implémentable

2.1 Les algorithmes de type min-max semi-infini.

Au pas 1 de l'algorithme du type Dinkelbach (DTA), nous avons résolu exactement le sous-problème

$$(P_{\lambda_k}) \qquad \min_{x\in X}\left\{F(\lambda_k,x)=\max_{t\in T}\{f_t(x)-\lambda_k g_t(x)\}\right\}$$

et nous avons utilisé sa solution x^k pour construire l'itération suivante

$$\lambda_{k+1}=\max_{t\in T}\left\{f_t(x^k)/g_t(x^k)\right\}.$$

Dans cette section nous indiquerons comment résoudre le problème de min-max semi-infini. Nous procéderons en deux étapes. D'abord nous présenterons rapidement la méthode classique d'approximation externe et ensuite nous expliquerons comment approximer une fonction maximum non différentiable par une fonction différentiable. Ceci nous permettra de construire un algorithme implémentable pour résoudre un problème général de min-max semi-infini.

2.1.1 La méthode d'approximation externe.

Considérons le problème de min-max semi-infini

$$(MMP) \qquad \min_{x\in X}\left\{\psi(x)=\max_{t\in T}\phi_t(x)\right\}$$

où X est un sous-ensemble compact de $\mathbb{R}^n$, T est un espace métrique compact et $\phi_t(x)$ est continue sur $X \times T$.

Dans les méthodes d'approximation externe, la stratégie est de construire une suite $\{T_m\}_{m\in\mathbb{N}}$ de sous-ensembles finis de T tel que chaque point limite de la suite $\{x^m\}$ définie pour chaque m par

$$x^m \cong \arg\min_{x\in X} \psi_{T_m}(x)$$

soit un minimum global de ψ sur X.

Le plus simple exemple de la méthode conceptuelle d'approximation externe pour résoudre le problème (MMP) a la forme ci-dessous, où nous utilisons la notation suivante

$$T(x) = \arg\max_{t\in T} \phi_t(x).$$

L'algorithme MOA

PAS 0 : Choisir un $x^0 \in X$, poser $m = 0$, prendre $t_0 \in T(x^0)$ et poser $T_0 = \{t_0\}$.

PAS 1 : Prendre $x^{m+1} \in \arg\min_{x\in X} \psi_{T_m}(x)$.

PAS 2 : Prendre un $t_{m+1} \in T(x^{m+1})$, et poser $T_{m+1} = T_m \cup \{t_{m+1}\}$.

PAS 3 : Remplacer m par $m+1$, et retourner au pas 1.

Notons que l'agorithme conceptuel est bien défini puisque X étant compact, chaque sous-problème $\min_{x\in X} \psi_{T_m}(x)$ a une solution. Le simple résultat suivant est la clé de l'analyse des méthodes d'approximation externe.

Lemme 13.
Soit $\{x^m\}$ une suite infinie générée par l'algorithme MOA. Si $\bar{x}$ est un point limite de $\{x^m\}$, tel que pour un sous-ensemble infini $K \subseteq \mathbb{N}$, $x^m \to_K \bar{x}$ lorsque $m \to \infty$, alors

$$\psi_{T_m}(x^m) \to_K \psi_T(\bar{x}) = \psi(\bar{x}), \text{ lorsque } m \to \infty.$$

Preuve.

Soit $x^m \to_K \bar{x}$, lorsque $m \to \infty$, pour un sous-ensemble infini $K \subseteq \mathbb{N}$. Pour n'importe quel $m \in \mathbb{N}$, prenons un $t_m \in T(x^m)$ comme dans le pas 2 de l'algorithme MOA, et pour n'importe quel $m \in K$, posons $k(m) = \max\{m' \in K \mid m' < m\}$. Alors, par construction, pour n'importe quel $m \in K$, $t_{k(m)} \in T_m$ et donc, comme $T_m \subseteq T$, pour n'importe quel $m \in K$,

$$\psi_T(x^m) = \max_{t \in T} \phi_t(x^m) \geq \max_{t \in T_m} \phi_t(x^m) = \psi_{T_m}(x^m) \geq \phi_{t_{k(m)}}(x^m). \tag{2.1}$$

Ensuite, puisque ψ_T est continu, $\psi_T(x^{k(m)}) \to_K \psi_T(\bar{x})$, lorsque $m \to \infty$. La fonction ϕ étant uniformément continue sur l'ensemble compact $X \times T$, et $\|(x^m, t_{k(m)}) - (x^{k(m)}, t_{k(m)})\| \to 0$ lorsque $m \to \infty$,

$$\phi_{t_{k(m)}}(x^m) - \phi_{t_{k(m)}}(x^{k(m)}) \to_K 0, \tag{2.2}$$

lorsque $m \to \infty$. Alors, puisque $\phi_{t_{k(m)}}(x^{k(m)}) = \psi_T(x^{k(m)}) \to_K \psi_T(\bar{x})$, lorsque $m \to \infty$, on a par (2.2) que $\phi_{t_{k(m)}}(x^m) \to_K \psi_T(\bar{x})$, lorsque $m \to \infty$. Il suit alors directement par le (2.1) que $\psi_{T_m}(x^m) \to_K \psi_T(\bar{x}) = \psi(\bar{x})$, lorsque $m \to \infty$. □

Proposition 14.
Soit $\{x^m\}$ une suite infinie générée par l'algorithme MOA. Si $\bar{x}$ est un point limite de $\{x^m\}$, alors $\bar{x}$ est un minimum global de ψ.

Preuve.

Par le lemme 13, on a que $\psi_{T_m}(x^m) \to_K \psi(\bar{x})$, lorsque $m \to \infty$. Soit $\bar{\alpha} = \min_{x \in X} \psi(x)$. Comme $T_m \subseteq T \ \forall m \in \mathbb{N}$, et que par construction $x^m \in \arg\min_{x \in X} \psi_{T_m}(x)$

$$\text{on a alors} \quad \forall x \in X \qquad \psi_{T_m}(x^m) \leq \psi_{T_m}(x) \leq \psi(x)$$

et par conséquent, $\psi_{T_m}(x^m) \leq \bar{\alpha} \ \forall m \in \mathbb{N}$, $\psi(\bar{x}) \leq \bar{\alpha}$. Or $\psi(\bar{x}) < \bar{\alpha}$ est impossible, donc nous avons que $\psi(\bar{x}) = \bar{\alpha}$, ce qui prouve que $\bar{x}$ est un minimum global de ψ sur X. □

Corollaire 15.

Si pour un certain $\hat{m} \in \mathbb{N}$, $t_{\hat{m}+1} \in T_{\hat{m}}$, alors la suite $\{x^m\}$ est constante pour $m \geq \hat{m}+1$ et $\bar{x} = x^{\hat{m}+1}$ est un minimum global de ψ sur X.

Preuve.

En effet, $t_{\hat{m}+1} \in T_{\hat{m}} \Rightarrow T_{\hat{m}+1} = T_{\hat{m}}$.

D'où $x^{\hat{m}+2} \in \arg\min_{x \in X} \psi_{T_{\hat{m}+1}} = \arg\min_{x \in X} \psi_{T_{\hat{m}}} \Rightarrow x^{\hat{m}+2} = x^{\hat{m}+1}$. En continuant de la même façon on obtient que $x^{\hat{m}+1} = x^{\hat{m}+2} = x^{\hat{m}+3} = x^{\hat{m}+4} = \ldots$ c'est-à-dire que $\bar{x} = x^{\hat{m}+1}$ est le point limite de la suite $\{x^m\}$ et donc par la proposition (14), que $\bar{x}$ est un minimum global de ψ sur X. □

2.1.2 Régularisation entropique.

Au pas 1 de l'algorithme conceptuel (MOA), nous avons résolu un problème fini du type min-max de la forme

$$\min_{x \in X} \left\{ \psi^m(x) = \max_{1 \leq i \leq m} \phi_{t_i}(x) \right\}.$$

Comme la fonction ψ^m est non différentiable en x, la stratégie sera d'abord de l'approximer par une fonction différentiable ψ^m_ϵ dépendant du paramètre ϵ et ensuite de minimiser cette fonction différentiable pour obtenir x^m. Dans le but de conserver la convergence de la méthode, nous avons besoin de connaître une estimation de l'erreur uniforme sur la différence entre ψ^m et ψ^m_ϵ et d'imposer que $\forall x \in X$, on ait que $\lim_{\epsilon \to 0^+} \psi^m_\epsilon(x) = \psi^m(x)$.

Dans cette section nous présenterons le concept de régularisation entropique pour approximer la fonction maximum ψ^m par une fonction différentiable. Ce concept est basé sur la propriété suivante.

Proposition 16.

Pour une fonction convexe $g : \mathbb{R}^m \to \mathbb{R}$, la fonction G définie par $G(x) = \lim_{\epsilon \to 0^+} \epsilon g(\epsilon^{-1}x)$ est bien définie. De plus, $\forall x \in \mathbb{R}^m$ et $\epsilon > 0$, on a l'inégalité

$$\epsilon g(\frac{x}{\epsilon}) - G(x) \leq \epsilon g(0).$$

Preuve.

(a) Montrons que la fonction $u \to u^{-1}\left[g(ux) - g(0)\right]$ est croissante sur $(0, \infty)$ c'est-à-dire que pour $s \geq u$,

$$\frac{g(sx) - g(0)}{s} \geq \frac{g(ux) - g(0)}{u}.$$

Soit la fonction $h : x \to h(x) = g(x) - g(0)$. Cette fonction est évidemment convexe puisque g est convexe. Remarquons que $\frac{g(sx)-g(0)}{s} = \frac{h(sx)}{s}$.

Soit $0 < u \leq s$,

$$\begin{aligned} h(ux) &= h\left[\tfrac{u}{s}sx + (1 - \tfrac{u}{s})0\right] \\ &\leq \tfrac{u}{s}h(sx) + (1 - \tfrac{u}{s})\underbrace{h(0)}_{=0} \\ &= \tfrac{u}{s}h(sx) \end{aligned}$$

$$\Longrightarrow \frac{h(ux)}{u} \leq \frac{h(sx)}{s} \Longrightarrow \frac{g(ux) - g(0)}{u} \leq \frac{g(sx) - g(0)}{s}.$$

(b) La fonction $u \to u^{-1}\left[g(ux) - g(0)\right]$ étant croissante sur $(0, \infty)$, la fonction $\epsilon \to \epsilon\left[g(\epsilon^{-1}x) - g(0)\right]$ est décroissante sur $(0, \infty)$. Nous avons donc que

$$\sup_{\epsilon>0} \epsilon\left[g(\frac{x}{\epsilon}) - g(0)\right] = \lim_{\epsilon\to 0^+} \epsilon\left[g(\frac{x}{\epsilon}) - g(0)\right] = \lim_{\epsilon\to 0^+} \epsilon g(\frac{x}{\epsilon}) = G(x),$$

$$\Longrightarrow G(x) \geq \epsilon\left[g(\frac{x}{\epsilon}) - g(0)\right].$$

□

Maintenant introduisons la fonction de type entropique $g : \mathbb{R}^m \to \mathbb{R}$ définie par

$$g(x) = \ln\left(\sum_{i=1}^{m} e^{x_i}\right).$$

Cette fonction est convexe et satisfait la propriété suivante :

$$\lim_{\epsilon\to 0^+} \epsilon g\left(\frac{x}{\epsilon}\right) = \lim_{\epsilon\to 0^+} \epsilon \ln\left(\sum_{i=1}^{m} e^{x_i/\epsilon}\right) = \max\left\{x_1, \cdots, x_m\right\}.$$

En effet, soit $x_j = \max\{x_1, \cdots, x_m\}$. Alors, pour $\epsilon > 0$, nous avons que

$$x_j = \epsilon \ln e^{x_j/\epsilon} \leq \epsilon \ln \left(\sum_{i=1}^{m} e^{x_i/\epsilon} \right) \leq \epsilon \ln(m e^{x_j/\epsilon})$$
$$= \epsilon \left(\ln m + \ln e^{x_j/\epsilon}\right) = \epsilon \left(\ln m + \frac{x_j}{\epsilon} \ln e\right) = \epsilon \ln m + x_j$$

En passant à la limite on obtient alors

$$x_j \leq \lim_{\epsilon \to 0^+} \epsilon \ln \left(\sum_{i=1}^{m} e^{x_i/\epsilon} \right) \leq x_j,$$

$$\Longrightarrow \lim_{\epsilon \to 0^+} \epsilon \ln \left(\sum_{i=1}^{m} e^{x_i/\epsilon} \right) = x_j = \max\{x_1, \cdots, x_m\}.$$

En conséquence, la fonction $G(x) = \max\{x_1, \cdots, x_m\}$ est approximée par la fonction

$$G_\epsilon(x) = \epsilon \ln \left(\sum_{i=1}^{m} e^{x_i/\epsilon} \right)$$

avec la propriété :

$$0 \leq G_\epsilon(x) - G(x) \leq \epsilon \ln m \qquad \forall x \in \mathbb{R}^m.$$

En particulier, la fonction $\psi^m(x) \equiv \max_{1 \leq i \leq m} \{\phi_{t_i}(x)\}$ est approximée par la fonction

$$\psi^m_\epsilon(x) = \epsilon \ln \left(\sum_{i=1}^{m} e^{\phi_{t_i}/\epsilon} \right)$$

avec la propriété que $\forall x \in X$

$$0 \leq \psi^m_\epsilon(x) - \psi^m(x) \leq \epsilon \ln m.$$

Tel que $\psi^m_\epsilon(x) \to \psi^m(x)$ lorsque $\epsilon \to 0^+$. De plus, si les fonctions ϕ_{t_i}, $i = 1, \cdots, m$ sont de classes C^1, alors la fonction $\psi^m_\epsilon(x)$ est aussi de classe C^1 et les méthodes de type quasi-Newton peuvent être utilisées pour la minimiser.

2.1.3 Algorithme de régularisation entropique itérative.

Dans cette section nous proposons une méthode d'approximation externe pour résoudre le problème continu de min-max semi-infini

$$(MMP) \qquad \min_{x \in X} \left\{ \psi(x) = \max_{t \in T} \phi_t(x) \right\}$$

où X est un sous-ensemble compact de $\mathbb{R}^n$, T est un espace métrique compact et $\phi_t(x)$ est continu sur $X \times T$.

Puisque nous utilisons la méthode d'approximation externe, l'ensemble compact infini T est approximé par un sous-ensemble fini T_m de m points de T. Réciproquement la fonction $\psi(x)$ est approximée par la fonction

$$\psi^m(x) = \psi_{T_m}(x) = \max_{t \in T_m} \{\phi_t(x)\}, \qquad \forall x \in X.$$

Comme ψ^m est non différentiable, nous considérons sa régularisée entropique ψ_ϵ^m, $\epsilon > 0$ qui satisfait la propriété

$$\lim_{\epsilon \to 0^+} \psi_\epsilon^m(x) = \psi^m(x), \qquad \forall x \in X.$$

Rappelons-nous que $\forall x \in X$,

$$\psi_\epsilon^m(x) = \epsilon \ln \left\{ \sum_{t \in T_m} e^{\phi_t(x)/\epsilon} \right\},$$

qui admet une estimation de l'erreur uniforme bien connue

$$0 \leq \psi_\epsilon^m(x) - \psi^m(x) \leq \epsilon \ln m. \tag{2.3}$$

L'algorithme IER

PAS 1 : Prendre $t_1 \in T$ et poser $T_1 = \{t_1\}$, $m = j = 1$. Choisir $\delta \in (0,1)$, $\epsilon > 0$.

PAS 2 : Trouver $x_\epsilon^m \in X$ satistaisant

$$\psi_\epsilon^m(x_\epsilon^m) \leq \min_{x \in X} \psi_\epsilon^m(x) + \delta^j. \tag{2.4}$$

Augmenter l'indice d'itération j par 1.

PAS 3 : Si (i) $\psi(x_\epsilon^m) \leq \psi_\epsilon^m(x_\epsilon^m)$ et (ii) $\delta^j + \epsilon \ln m$ est en dessous d'une tolérance désirée, stop. Si (i) est violée, alors choisir un $t_{m+1} \in \arg\max_{t \in T} \phi_{t \in T}(x_\epsilon^m)$, poser $T_{m+1} = T_m \cup \{t_{m+1}\}$, augmenter m par 1, choisir $0 < \epsilon \leq (\ln m)^{-2}$, et retourner au pas 2. Si (ii) est violée, diviser ϵ par un facteur constant, et retourner au pas 2.

Observons que la condition (i) au pas 3 signifie que ψ_ϵ^m est une bonne approximation de ψ autour de x_ϵ^m lorsque $\epsilon \ln m$ est petit. En effet, si (i) est satisfait, nous avons en utilisant successivement (2.3) et la définition de ψ^m,

$$\psi(x_\epsilon^m) \leq \psi_\epsilon^m(x_\epsilon^m) \leq \psi^m(x_\epsilon^m) + \epsilon \ln m \leq \psi(x_\epsilon^m) + \epsilon \ln m.$$

Observons aussi que le stop au pas 3 implique que x_ϵ^m est une solution optimale à $\delta^j + \epsilon \ln m$ près.

Proposition 17.
L'algorithme IER, soit stop au pas 3 avec une solution optimale à une précision de $\delta^j + \epsilon \ln m$ près, ou soit, génère une suite infinie $\{x_\epsilon^m\} \in X$ $(m \to \infty,\ \epsilon \leq (\ln m)^{-2})$, dont n'importe quel point limite est un minimum global du problème (MMP).

Preuve.
Supposons que (i) du pas 3 est violée un nombre fini de fois. Alors, m est fixé par exemple à un certain $\bar{m}$ et $\epsilon \to 0,\ j \to \infty$, tels que $\epsilon \log \bar{m} \to 0$ et $\delta^j \to 0$.

$$\begin{aligned} \psi(x_\epsilon^{\bar{m}}) &\overset{(i)}{\leq} \psi_\epsilon^{\bar{m}}(x_\epsilon^{\bar{m}}) \overset{(2.4)}{\leq} \psi_\epsilon^{\bar{m}}(x^*) + \delta^j \\ &\overset{(12)}{\leq} \psi^{\bar{m}}(x^*) + \delta^j + \epsilon \ln \bar{m} \leq \psi(x^*) + \delta^j + \epsilon \ln \bar{m}, \end{aligned}$$

où x^* est une solution optimale du problème (MMP). Alors, $x_\epsilon^{\bar{m}}$ est une solution optimale à $\delta^j + \epsilon \ln \bar{m}$ près.
Suposons que (i) du pas 3 est violée indéfiniment. Si $m \to \infty$, alors $\epsilon \leq (\ln m)^{-2}$ implique que $\epsilon \log m \to 0$ ainsi que $\delta^j \to 0$.

$$\begin{aligned} \psi^m(x_\epsilon^m) &\overset{(2.3)}{\leq} \psi_\epsilon^m(x_\epsilon^m) + \epsilon \ln m \overset{2.4)}{\leq} \psi_\epsilon^m(x) + \delta^j + \epsilon \ln m \\ &\overset{(2.3)}{\leq} \psi^m(x) + \delta^j + 2\epsilon \ln m \leq \psi(x) + \delta^j + 2\epsilon \ln m, \end{aligned} \tag{2.5}$$

$\forall x \in X$. Soit $(\bar{x}, \bar{t})$ un point limite de la suite $\{(x_\epsilon^m, t_{m+1})\}$ lorsque $m \to \infty$ avec $\epsilon \leq (\ln m)^{-2}$. Alors, il existe un ensemble infini $M \subseteq \mathbb{N}_0$ tel que, lorsque $m \to \infty,\ m \in M$, nous avons que $\{(x_\epsilon^m, t_{m+1})\}$ converge vers $(\bar{x}, \bar{t})$ dans l'espace compact $X \times T$. Alors,

$$\phi_{t_{m+1}}(x_\epsilon^m) = \psi(x_\epsilon^m) \geq \psi^m(x_\epsilon^m) \geq \phi_{t_{i(m)+1}}(x_\epsilon^m), \tag{2.6}$$

où $i(m) = \max\{i \mid i \in M \cap \{1, ..., m-1\}\}$ pour $m > 1$.
Alors, $t_{i(m)+1} \to \bar{t}$ lorsque $m \to \infty,\ m \in M$.
Par la continuité de $\phi_t(x)$, les deux côtés de (2.7) tendent vers $\phi_{\bar{t}}(x)$ ce qui implique que $\psi^m(x_\epsilon^m) \to \psi(x_\epsilon^m)$. Or, $\psi(x_\epsilon^m) \to \psi(\bar{x})$ car ψ est continue. En utilisant ceci et en prenant la limite dans (2.5), nous avons que

$$\psi(\bar{x}) \leq \psi(x), \qquad \forall x \in X,$$

d'où $\bar{x}$ est un minimum global du problème (MMP). □

Proposition 18.
Si l'algorithme IER génère une suite infinie $\{x_\epsilon^m\}$. Alors, $\psi_\epsilon^m(x_\epsilon^m) \to \psi(\bar{x})$ lorsque $m \to \infty$, où $\bar{x}$ est un minimum global du problème (MMP).

Preuve.

$\forall x \in X$, $\{\psi^m(x)\}_{m=1}^{\infty}$ forme une suite croissante bornée supérieurement par $\psi(x)$.

Définissons $\bar{\psi}(x) = \lim_{m\to\infty} \psi^m(x) \quad \forall x \in X$.

Théorème de Dini : " Une suite croissante de fonctions continues à valeurs réelles $\{\psi^m\}_{m=1}^{\infty}$ qui converge points par points vers une fonction continue f sur un espace métrique compact converge uniformément vers cette fonction f".

Remarquons que $\bar{\psi}$ est continue sur X. Alors, si l'algorithme génère une suite infinie $\{x_\epsilon^m\}$, par l'algorithme de Dini, $\{\psi^m\}_{m=1}^{\infty}$ converge uniformément vers $\bar{\psi}$.

1) Montrons que $\min_{x\in X} \psi(x) = \min_{x\in X} \bar{\psi}(x)$. Par l'équation (2.5), nous avons

$$\psi^m(x_\epsilon^m) \leq \psi^m(x) + \delta^j + 2\epsilon \ln m.$$

Passons à la limite lorsque $m \to \infty$, alors par le théorème 17, $\psi(\bar{x}) \leq \bar{\psi}(x)$.

Or, $\bar{\psi}(x) \leq \psi(x)$ $\forall x \in X$, et donc en particulier pour $\bar{x}$ on a que $\bar{\psi}(\bar{x}) \leq \psi(\bar{x})$.

$$\Rightarrow \bar{\psi}(\bar{x}) \leq \bar{\psi}(x),$$

c'est-à-dire que $\bar{x}$ minimise $\bar{\psi}$.

2) Montrons que $\psi_\epsilon^m(x_\epsilon^m) \to \psi(\bar{x})$.

$$\begin{aligned} 0 &\leq \left|\psi_\epsilon^m(x_\epsilon^m) - \bar{\psi}(\bar{x})\right| \\ &= \left|\psi_\epsilon^m(x_\epsilon^m) - \psi^m(x_\epsilon^m) + \psi^m(x_\epsilon^m) - \bar{\psi}(x_\epsilon^m) + \bar{\psi}(x_\epsilon^m) - \bar{\psi}(\bar{x})\right| \\ &\leq \underbrace{\left|\psi_\epsilon^m(x_\epsilon^m) - \psi^m(x_\epsilon^m)\right|}_{(a)} + \underbrace{\left|\psi^m(x_\epsilon^m) - \bar{\psi}(x_\epsilon^m)\right|}_{(b)} + \underbrace{\left|\bar{\psi}(x_\epsilon^m) - \bar{\psi}(\bar{x})\right|}_{(c)} \end{aligned}$$

$(a) \leq |\epsilon \ln m|$ par (2.3)

Lorsque $m \to +\infty$, on a que

$(a) \to 0$,

$(b) \to 0$ par convergence uniforme de ψ^m vers $\bar{\psi}$,

$(c) \to 0$ par continuité de $\bar{\psi}$,

et donc $\left|\psi_\epsilon^m(x_\epsilon^m) - \bar{\psi}(\bar{x})\right| \to 0$ ce qui implique le résultat recherché. □

2.2 Comment rendre implémentable l'algorithme de Dinkelbach.

Dans cette section nous adapterons l'algorithme de régularisation entropique pour résoudre le problème fractionnel

$$(P) \qquad \lambda^* = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)}{g_t(x)} \right\} \right\},$$

où X est un sous-ensemble compact non vide de $\mathbb{R}^n$, T est un espace métrique compact, $f_t(x)$, $g_t(x)$ sont des fonctions continues sur $T \times X$, et $g_t > 0$ sur X $\forall t \in T$. En particulier, notre but est de rendre implémentable l'algorithme de type Dinkelbach (DTA) rencontré dans la section 3 du chapitre 1. Rappelons que dans le pas 1 de cet algorithme le sous-problème de type min-max

$$(P_{\lambda_k}) \qquad \min_{x \in X} \left\{ F(\lambda_k, x) = \max_{t \in T} \{ f_t(x) - \lambda_k g_t(x) \} \right\}$$

est résolu exactement et sa solution x^k est utilisée pour construire l'itération suivante

$$\lambda_{k+1} = \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\}.$$

Dans le but d'appliquer l'algorithme IER pour résoudre le problème (P_{λ_k}), nous introduisons les notations suivantes : soit

$$F_\lambda(x) = \max_{t \in T} \{ f_t(x) - \lambda g_t(x) \}.$$

Puisque T est infini, T est approximé par un sous-ensemble fini T_m de m points dans T. Réciproquement, $F_\lambda(x)$ est approximée par la fonction

$$F_\lambda^m(x) = \max_{t \in T_m} \{ f_t(x) - \lambda g_t(x) \}, \qquad \forall x \in X.$$

Comme F_λ^m est non différentiable, nous considérons sa régularisée entropique $F_{\epsilon,\lambda}^m$, $\epsilon > 0$, qui satisfait la propriété

$$\lim_{\epsilon \to 0^+} F_{\epsilon,\lambda}^m(x) = F_\lambda^m(x), \qquad \forall x \in X.$$

Rappelons-nous que, $\forall \lambda \in \mathbb{R}$ et $x \in X$,

$$F_{\epsilon,\lambda}^m(x) = \epsilon \ln \left(\sum_{t \in T_m} e^{(f_t(x) - \lambda g_t(x))/\epsilon} \right)$$

avec la propriété que $\forall \lambda \in \mathbb{R}$, $x \in X$,

$$0 \leq F_{\epsilon,\lambda}^m(x) - F_\lambda^m(x) \leq \epsilon \ln m. \tag{2.7}$$

De plus, si les fonctions f_t et g_t, $t \in T_m$ sont de classe C^1, alors la fonction $F^m_{\epsilon,\lambda}(x)$ est aussi de classe C^1 et les méthodes quasi-Newton sont dès lors applicables pour la minimiser. Nous pouvons alors utiliser l'algorithme IER pour résoudre exactement chaque sous-problème (P_{λ_k}). L'algorithme de type Dinkelbach résultant devient implémentable mais très coûteux d'un point de vue numérique. Heureusement, on peut observer que parfois ce n'est pas la meilleure stratégie. En effet, si à la place de résoudre exactement (P_{λ_k}) pour obtenir x^k et λ_{k+1}, nous choisissons un vecteur différent $\bar{x}^k$ satisfaisant $F(\lambda_k, \bar{x}^k) < 0$, alors il se peut que

$$\lambda^*_{k+1} = \max_{t \in T} \left\{ f_t(\bar{x}^k)/g_t(\bar{x}^k) \right\},$$

le paramètre associé avec $\bar{x}^k$, soit plus proche de λ^* que λ_{k+1} obtenu à partir de x^k.

Exemple : supposons que nous voulons résoudre

$$\lambda^* = \min_{0 \le x \le 2} \max \left\{ \frac{-7x+1}{2x+2}, \ \frac{-18x+2}{4x+1}, \ \frac{3x-2}{16x+3} \right\}.$$

Soit $x^0 = 1$. Alors $\lambda_1 = 0.5263$. Observons que le problème (P_{λ_1}) peut être résolu par un solver de programmation linéaire standard. Après avoir résolu le problème (P_{λ_1}), nous trouvons la solution optimale $x^1 = 0.3295$ avec la fonction objective qui a la valeur $F(\lambda_1, x^1) = -1.4468$. A l'itération suivante nous calculons $\lambda_2 = 0.1228$. D'un autre côté, à la place de la solution optimale, si nous choisissons $\bar{x}^1 = 0.30$, ça nous conduit à $F(\lambda_1, \bar{x}^1) = -1.2368$, alors la valeur correspondante de λ à l'itération suivante devient $\lambda^*_2 = -0.1410$, ce qui est plus petit que λ_2.

Ceci suggère qu'investir du temps à résoudre le problème (P_{λ_k}) de façon optimale n'est pas toujours la meilleure stratégie. En particulier, cette observation peut devenir utile quand la routine d'optimisation pour résoudre le problème auxiliaire exige un effort considérable. Ainsi notre stratégie ici ne sera pas de résoudre exactement chaque sous-problème (P_{λ_k}) mais de trouver un point $x^m_{\epsilon,\lambda_k} \in X$ tel que

$$F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) \le F^m_{\epsilon,\lambda_k}(x) + \delta^j, \qquad \forall x \in X. \tag{2.8}$$

où $0 < \delta < 1$ et $j \in \mathbb{N}_0$.

La difficulté principale est comment choisir et mettre à jour les paramètres ϵ, δ et les sous-ensembles T_m de T pour obtenir la convergence de la suite $\{\lambda_k\}$ vers la valeur optimale λ^* de (P). Pour répondre à cette question, d'abord nous pouvons observer que si $x^k \in X$ satisfait $F_{\lambda_k}(x^k) < 0$ et est utilisée (à la place du point optimal de (P_{λ_k})) pour mettre à jour la valeur de λ_k :

$$\lambda_{k+1} = \max_{t \in T} \left\{ f_t(x^k)/g_t(x^k) \right\},$$

alors

$$\lambda_{k+1} - \lambda_k = \frac{f_{t^*}(x^k) - \lambda_k g_{t^*}(x^k)}{g_{t^*}(x^k)} \leq \frac{F(\lambda_k, x^k)}{g^*}$$

où t^* est le point de T donnant le maximum dans l'expression de λ_{k+1}, et $g^* = \max_{t \in T} \max_{x \in X} g_t(x)$. Donc, si nous assurons que $g^* < \infty$ et $F(\lambda_k, x^k) < -\eta$ avec $\eta > 0$, alors

$$\lambda_k - \lambda_{k+1} > \frac{\eta}{g^*}.$$

Dans ce cas nous obtenons une décroissance fixée de λ_k qui semble intéressante pour converger aussi rapidement que possible vers la valeur optimale λ^* de (P). Pour résumer, dès qu'un point $x^m_{\epsilon,\lambda_k} \in X$ est obtenu tel que (2.8) et $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) < -\eta$ sont satisfaits, le paramètre λ_k est mis à jour peu importe que x^m_{ϵ,λ_k} soit une bonne approximation ou pas de la solution optimale du sous-problème (P_{λ_k}). D'un autre côté, si $x^m_{\epsilon,\lambda_k} \in X$ satisfait (2.8) mais que $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \geq -\eta$, alors nous testons si x^m_{ϵ,λ_k} est une bonne approximation de la solution optimale du sous-problème (P_{λ_k}) et nous mettons à jour λ_k seulement si c'est le cas. Comme dans l'algorithme IER, nous considérons que l'approximation est bonne si les deux conditions suivantes sont satisfaites :

$$F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) \quad \text{et} \quad \delta^j + \epsilon \ln m < \gamma,$$

où $\gamma > 0$ est le paramètre de tolérance.

Si la première condition n'est pas satisfaite, alors nous ajoutons un point t_{m+1} à T et nous améliorons la précision en prenant $\epsilon \leq (\ln m)^{-2}$ avant de calculer un nouveau point x^m_{ϵ,λ_k}. D'un autre côté, si la seconde condition n'est pas satisfaite, on diminue ϵ par un facteur constant avant de calculer un nouveau point x^m_{ϵ,λ_k}. Finallement, pour éviter le mauvais conditionnement, le paramètre $\epsilon > 0$ n'est pas choisi trop petit à la première itération pour être diminué si nécessaire après chaque itération. L'algorithme correspondant s'appelle MDER.

L'algorithme MDER

PAS 1 : Soient $x^0 \in X$, $t_1 \in T$, et $\epsilon, \eta, \gamma > 0$, $0 < \delta < 1$ donnés, calculer $\lambda_1 = \max_{t \in T} \{f_t(x^0)/g_t(x^0)\}$. Poser $m = 1$, $T_1 = \{t_1\}$, $k = 1$, $j = 1$, $x^m_{\epsilon,\lambda_k} = x^0$.

PAS 2 : Prendre $x^m_{\epsilon,\lambda_k} \in X$ qui satisfait

$$F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F^m_{\epsilon,\lambda_k}(x) + \delta^j, \qquad \forall x \in X,$$

et calculer $F_{\lambda_k}(x^m_{\epsilon,\lambda_k})$.

PAS 3 : Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) < -\eta$, mettre à jour le paramètre

$$\lambda_{k+1} = \max_{t \in T} \{f_t(x^m_{\epsilon,\lambda_k})/g_t(x^m_{\epsilon,\lambda_k})\};$$

poser $k = k + 1$, et retourner au pas 2.

PAS 4 : Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \geq -\eta$, améliorer si nécessaire, m et ϵ pour une meilleure approximation.

PAS 4.1 : Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) > F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) + \delta_k$, prendre

$$t_{m+1} \in \arg\max_{t \in T} \{f_t(x^m_{\epsilon,\lambda_k}) - \lambda_k g_t(x^m_{\epsilon,\lambda_k})\},$$

Diviser ϵ par un facteur constant, poser $T_{m+1} = T_m \cup \{t_{m+1}\}$, $j = j + 1$, et $m = m + 1$; retourner au pas 2.

PAS 4.2 : Si $\delta^j + \epsilon \ln m > \gamma - \delta_k$, diviser ϵ par un facteur constant, poser $j = j + 1$ et retourner au pas 2.

PAS 5 : Si $\left|F_{\lambda_k}(x^m_{\epsilon,\lambda_k})\right| > \gamma$, calculer

$$\lambda_{k+1} = \max_{t \in T} \{f_t(x^m_{\epsilon,\lambda_k})/g_t(x^m_{\epsilon,\lambda_k})\},$$

poser $k = k + 1$, $j = 1$, et retourner au pas 2. Sinon, stopper et rendre x^m_{ϵ,λ_k} comme solution optimale à γ près du problème (P).

D'abord observons qu'il n'y a pas de boucle infinie entre le pas 2 et le pas 3, ni entre le pas 2 et le pas 4. En effet, nous avons vu que $\lambda_k - \lambda_{k+1} \geq \eta/g^*$ lorsque $F_{\lambda_k}(x_\epsilon^m \lambda_k) < -\eta$. Il est donc impossible d'avoir une boucle infinie entre le pas 2 et le pas 3 puisque λ^* est fini par la proposition (1). Maintenant, dans le cas d'une boucle infinie entre le pas 2 et le pas 4, nous pouvons observer que, k étant fixé, l'inégalité suivante

$$F_{\lambda_k}(x_{\epsilon,\lambda}^m) > F_{\epsilon,\lambda_k}^m(x_{\epsilon,\lambda}^m) + \delta^k, \tag{2.9}$$

tient pour m assez grand.

Utilisant les résultats de convergence de l'algorithme IER pour résoudre le problème de min-max (P_{λ_k}), nous obtenons du théorème (17), que $F_{\epsilon,\lambda_k}^m(x_{\epsilon,\lambda}^m) \to F_{\lambda_k}(x_{\lambda_k}^*)$, et du théorème (18), que $F_{\epsilon,\lambda_k}^m(x_{\epsilon,\lambda_k}^m) \to F_{\lambda_k}(x_{\lambda_k}^*)$, $\forall m$, où $x_{\lambda_k}^*$ dénote le minimum de F_{λ_k} sur X. En prenant la limite sur m dans (2.9), nous déduisons que

$$F_{\lambda_k}(x_{\lambda_k}^*) \geq F_{\lambda_k}(x_{\lambda_k}^*) + \delta^k,$$

ce qui est impossible puisque δ^k est fixé strictement positif. Il n'y a donc pas de boucle infinie entre le pas 2 et le pas 4.

Concernant le critère d'arrêt, observons d'abord que $\forall k$, nous avons

$$\lambda_k = \max_{t \in T} \left\{ f_t(x_{\epsilon,\lambda_{k-1}}^m)/g_t(x_{\epsilon,\lambda_{k-1}}^m) \right\} \geq \min_{x \in X} \max_{t \in T} \{f_t(x)/g_t(x)\} = \lambda^*,$$

et, puisque $g_t > 0 \ \forall t \in t$,

$$F(\lambda_k) = \min_{x \in X} \max_{t \in T} [f_t(x) - \lambda_k g_t(x)] \leq \min_{x \in X} \max_{t \in T} [f_t(x) - \lambda^* g_t(x)] = F(\lambda^*) = 0.$$

Maintenant au travers d'une suite de lemmes, nous allons montrer que lorsque le critère d'arrêt $\left| F_{\lambda_k}(x_{\epsilon,\lambda_k}^m) \right| \leq \gamma$ est satisfait, $F(\lambda_k)$, la valeur optimale du problème (P_{λ_k}) est dans un voisinage 2γ de 0, ce qui veut dire que x_{ϵ,λ_k}^m est une approximation de la solution optimale du problème (P).

Lemme 19.

Soit T_m un sous-ensemble fini de T et $\epsilon > 0$. Alors,

$$F^m_{\epsilon,\lambda_k}(x^*_{\lambda_k}) \leq \max_{t\in T}\left[f_t(x^*_{\lambda_k}) - \lambda_k g_t(x^*_{\lambda_k})\right] + \epsilon \ln m,$$

où $x^*_{\lambda_k}$ est la solution optimale de (P_{λ_k}).

Preuve.

En substituant $x^*_{\lambda_k}$ dans (2.7), nous obtenons

$$\begin{aligned} F^m_{\epsilon,\lambda_k}(x^*_{\lambda_k}) &\leq \max_{t\in T_m}\left[f_t(x^*_{\lambda_k}) - \lambda_k g_t(x^*_{\lambda_k})\right] + \epsilon \ln m \\ &\leq \max_{t\in T}\left[f_t(x^*_{\lambda_k}) - \lambda_k g_t(x^*_{\lambda_k})\right] + \epsilon \ln m. \end{aligned}$$

□

Lemme 20.

Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) + \delta^k$, alors

$$F(\lambda_k) \leq F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F(\lambda_k) + \epsilon \ln m + \delta^j + \delta^k.$$

Preuve.

Par la définition de $F(\lambda_k)$ et $F_{\lambda_k}(x^m_{\epsilon,\lambda_k})$,

$$\begin{aligned} F(\lambda_k) &= \min_{x\in X} \max_{t\in T}\left[f_t(x) - \lambda_k g_t(x)\right] \\ &\leq \max_{t\in T}\left[f_t(x^m_{\epsilon,\lambda_k}) - \lambda_k g_t(x^m_{\epsilon,\lambda_k})\right] \\ &= F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) + \delta^k \\ &\leq \min_{x\in X} F^m_{\epsilon,\lambda_k}(x) + \delta^j + \delta^k. \end{aligned}$$

Alors, par le lemme 19,

$$\begin{aligned} F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) &\leq \min_{x\in X} F^m_{\epsilon,\lambda_k}(x) + \delta^j + \delta^k \\ &\leq F^m_{\epsilon,\lambda_k}(x^*_{\lambda_k}) + \delta^j + \delta^k \\ &\leq \max_{t\in T}\left[f_t(x^\star_{\lambda_k}) - \lambda_k g_t(x^\star_{\lambda_k})\right] + \epsilon \ln m + \delta^j + \delta^k \\ &= F(\lambda_k) + \epsilon \ln m + \delta^j + \delta^k. \end{aligned}$$

□

Proposition 21.

Quand l'algorithme stop au pas 5 en k, nous avons

$$-2\gamma \leq F_{\lambda_k} \leq 0.$$

Preuve.

En k, les conditions du pas 4.1, du pas 4.2, et du pas 5 sont toutes violées. On a donc le système d'inégalités suivant :

$$\begin{array}{lcll} F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) & \leq & F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) + \delta^k, & \neg(4.1) \\ \delta^j + \epsilon \ln m & \leq & \gamma - \delta^k & \neg(4.2) \\ \left|F_{\lambda_k}(x^m_{\epsilon,\lambda_k})\right| & \leq & \gamma. & \neg(5) \end{array}$$

Par (4.1), on peut utiliser successivement le lemme 20 et $\neg(4.2)$, et obtenir

$$\begin{aligned} F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) &\leq F(\lambda_k) + \epsilon \ln m + \delta^j + \delta^k \\ &\leq F(\lambda_k) + \gamma - \delta^k + \delta^k. \end{aligned}$$

$$F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F(\lambda_k) + \gamma. \tag{2.10}$$

Finallement, par $\neg(5)$ et (2.10), nous avons $-\gamma \leq F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F(\lambda_k) + \gamma$, ce qui implique que

$$-2\gamma \leq F(\lambda_k) \leq 0.$$

□

Par le théorème 21, nous appelons x^m_{ϵ,λ_k} une solution optimale-2γ et λ_k une valeur optimale-2γ de (P).

Maintenant, en théorie, il y a un risque que l'algorithme ne s'arrête jamais. Chaque passage à travers le pas 5 crée trois suites infinies $\left\{F_{\lambda_k}(x^m_{\epsilon,\lambda_k})\right\}$, $\{\lambda_k\}$, $\left\{x^m_{\epsilon,\lambda_k}\right\}$.

Notons $m(g) = \min_{(t,x)\in T\times X} g_t(x)$ et $M(g) = \max_{(t,x)\in T\times X} g_t(x)$.

Proposition 22.

Si l'algorithme MDER ne s'arrête pas au pas 5, alors $-\gamma + F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F(\lambda_k) \leq 0$. *Dans ce cas, la suite* $\{\lambda_k\}$ *est décroissante, bornée inférieurement par* $\lambda^\star$ *et satisfait les inégalités*

$$F(\lambda_k) \leq (\lambda^\star - \lambda^k) \min_{t\in T} g_t(x^\star)$$

et

$$(\lambda_{k+1} - \lambda^\star)/(\lambda_k - \lambda^\star) \leq 1 - m(g)/M(g) - e_k,$$

où

$$e_k = F_{\lambda_k}(x^m_{\epsilon,\lambda_k})/\left[(\lambda_k - \lambda^\star) \max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})\right].$$

Preuve.

1) Montrons que $-\gamma + F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F(\lambda_k) \leq 0$

Pour passer dans le pas 5 de l'algorithme MDER, il faut que $\left|F_{\lambda_k}(x^m_{\epsilon,\lambda_k})\right| > \gamma$ (a), mais aussi ne pas satisfaire les conditions des pas précédents :

$\neg$ PAS 3 $\Rightarrow F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \geq -\eta \qquad (b)$

$\neg$ PAS 4.1 $\Rightarrow F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) + \delta^k \qquad (c)$

$\neg$ PAS 4.2 $\Rightarrow \delta^j + \epsilon \ln m \leq \gamma - \delta^k \qquad (d)$

Alors l'hypothèse (c) nous permet d'appliquer le lemme 20,

$$\begin{aligned} F_{\lambda_k}(x^m_\epsilon, \lambda_k) &\leq F(\lambda_k) + \epsilon \ln m + \delta^j + \delta^k \\ &\leq F(\lambda_k) + \gamma - \delta^k + \delta^k \\ &= F(\lambda_k) + \gamma \end{aligned}$$

La seconde inégalité est obtenue par (d).

2) Montrons que $\{\lambda_k\}$ est décroissante.

Par (a) on a que

$$F_{\lambda_k}(x_{\epsilon,\lambda_k}) < -\gamma < 0,$$

c'est-à-dire

$$\max_{t\in T}\{f_t(x_{\epsilon,\lambda_k}) - \lambda_k g_t(x_{\epsilon,\lambda_k})\} < 0,$$

et donc pour $\forall t \in T$,

$$f_t(x_{\epsilon,\lambda_k}) - \lambda_k g_t(x_{\epsilon,\lambda_k}) < 0.$$
$$\Rightarrow \lambda_k > \frac{f_t(x_{\epsilon,\lambda_k})}{g_t(x_{\epsilon,\lambda_k})} \quad \forall t \in T.$$

Alors, en particulier, pour un certain $\tilde{t} \in T$

$$\lambda_k \geq \lambda_{k+1} = \frac{f_{\tilde{t}}(x_{\epsilon,\lambda_k})}{g_{\tilde{t}}(x_{\epsilon,\lambda_k})}.$$

3) Montrons que $F(\lambda_k) \leq (\lambda^* - \lambda_k) \min_{t\in T} g_t(x^*)$.

Comme $\lambda^* - \lambda_k \leq 0$,

$$\begin{aligned} f_t(x) - \lambda_k g_t(x) &= f_t(x) - \lambda^* g_t(x) + (\lambda^* - \lambda_k) g_t(x), \\ &\leq f_t(x) - \lambda^* g_t(x) + (\lambda^* - \lambda_k) \min_{t\in T} g_t(x). \end{aligned}$$

En prenant le maximum sur T on obtient

$$\max_{t\in T} \{f_t(x) - \lambda_k g_t(x)\} \leq \max_{t\in T} \{f_t(x) - \lambda^* g_t(x)\} + (\lambda^* - \lambda_k) \min_{t\in T} g_t(x).$$

Et en particulier, pour x^*

$$\begin{aligned} \max_{t\in T} \{f_t(x^*) - \lambda_k g_t(x^*)\} &\leq \max_{t\in T} \{f_t(x^*) - \lambda^* g_t(x^*)\} + (\lambda^* - \lambda_k) \min_{t\in T} g_t(x^*), \\ &= F(\lambda^*) + (\lambda^* - \lambda_k) \min_{t\in T} g_t(x^*). \end{aligned}$$

Or par définition,

$$\begin{aligned} F(\lambda_k) &= \min_{x\in X} \max_{t\in T} \{f_t(x) - \lambda_k g_t(x)\}, \\ &\leq \max_{t\in T} \{f_t(x^*) - \lambda_k g_t(x^*)\}, \\ &\leq F(\lambda^*) + (\lambda^* - \lambda_k) \min_{t\in T} g_t(x^*), \\ &= (\lambda^* - \lambda_k) \min_{t\in T} g_t(x^*) \end{aligned}$$

.

4) Montrons que $(\lambda_{k+1} - \lambda^*)/(\lambda_k - \lambda^*) \leq 1 - \frac{m(g)}{M(g)} - \left(F_{\lambda_k}(x^m_{\epsilon,\lambda_k})/\left[(\lambda_k - \lambda^*) \max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})\right]\right)$.

Par (a) et par la définition de F_{λ_k}

$$-\gamma > F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \geq f_t(x^m_{\epsilon,\lambda_k}) - \lambda_k g_t(x^m_{\epsilon,\lambda_k}).$$

En particulier pour $\tilde{t} \in \arg\max f_t(x^m_{\epsilon,\lambda_k})/g_t(x^m_{\epsilon,\lambda_k})$,

$$\begin{aligned}
\lambda_k &\geq \frac{f_{\tilde{t}}(x^m_{\epsilon,\lambda_k})}{g_{\tilde{t}}(x^m_{\epsilon,\lambda_k})} + \frac{\gamma}{g_{\tilde{t}}(x^m_{\epsilon,\lambda_k})}, \\
&\geq \frac{f_{\tilde{t}}(x^m_{\epsilon,\lambda_k})}{g_{\tilde{t}}(x^m_{\epsilon,\lambda_k})} + \frac{\gamma}{\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})}, \\
&= \lambda_{k+1} + \frac{\gamma}{\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})}.
\end{aligned}$$

$$\begin{aligned}
\lambda_{k+1} &\leq \lambda_k - \gamma/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) \\
\lambda_{k+1} - \lambda^* &\leq \lambda_k - \lambda^* - \gamma/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) \\
\lambda_{k+1} - \lambda^* &= \lambda_{k+} - \lambda^* - \gamma/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) - F(\lambda_k)/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) + F(\lambda_k)/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) \\
\lambda_{k+1} - \lambda^* &= \lambda_{k+} - \lambda^* - [F(\lambda_k) + \gamma]/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) + F(\lambda_k)/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})
\end{aligned}$$

Par le point 3) de notre démonstration, on a que

$$\begin{aligned}
\lambda_{k+1} - \lambda^* &\leq \lambda_k - \lambda^* - [F(\lambda_k) + \gamma]/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) + [(\lambda^* - \lambda_k)\min_{t\in T} g_t(x^*)]/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) \\
&\leq \lambda_k - \lambda^* - [F(\lambda_k) + \gamma]/\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k}) + \left[(\lambda^* - \lambda^k)m(g)\right]/M(g) \\
\tfrac{\lambda_{k+1}-\lambda^*}{\lambda_k-\lambda^*} &\leq 1 - [F(\lambda_k) + \gamma]/\left[(\lambda_k - \lambda^*)\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})\right] - m(g)/M(g)
\end{aligned}$$

On conclut grâce à l'hypothèse (a) : $-(F(\lambda_k) + \gamma) \leq -F_{\lambda_k}(x^m_{\epsilon,\lambda_k})$,

$$\frac{\lambda_{k+1} - \lambda^*}{\lambda_k - \lambda^*} \leq 1 - F_{\lambda_k}(x^m_{\epsilon,\lambda_k})/\left[(\lambda_k - \lambda^*)\max_{t\in T} g_t(x^m_{\epsilon,\lambda_k})\right] - m(g)/M(g).$$

□

Remarquons que nous avons prouvé un résultat similaire mais avec $e_k = 0$ pour l'algorithme de Dinkelbach DTA (proposition 8). Dans ce cas nous obtenions la convergence linéaire de la suite $\{\lambda_k\}$ directement de ce résultat. Ici, pour obtenir la convergence linéaire nous devons imposer une condition technique.

Proposition 23.

Si l'algorithme MDER ne s'arrête pas au pas 5 et si on a la condition suivante

$$0 < -e_k < m(g)/M(g) - \gamma, \quad \forall k > K,$$

alors

$$(\lambda_{k+1} - \lambda^*)/(\lambda_k - \lambda^*) \leq 1 - \gamma, \quad \forall k > K.$$

En d'autres mots, la suite $\{\lambda_k\}$ converge linéairement vers λ^*.

Preuve.

En effet, si la condition est satisfaite, par la proposition 22 on a que

$$\frac{\lambda_{k+1} - \lambda^*}{\lambda_k - \lambda^*} \leq 1 - \frac{m(g)}{M(g)} - e_k \leq 1 - \frac{m(g)}{M(g)} + \frac{m(g)}{M(g)} - \gamma = 1 - \gamma.$$

□

Notons que comme pour l'algorithme IER, calculer certaines valeurs comme $F_{\lambda_k}(x^m_{\epsilon,\lambda_k})$ et t_{m+1} peut s'avérer être difficile car elles exigent de trouver un optimum global sur l'espace compact T. Une stratégie est de subdiviser T et de calculer le maximum sur cette division.

Chapitre 3

Codes de programmation en MATLAB.

3.1 Algorithme de Dinkelbach pour $T = \{1\}$.

L'algorithme de Dinkelbach pour $T = \{1\}$

PAS 0 : Soit $x^0 \in X$, $\lambda_1 = f(x^0)/g(x^0)$, et $k = 1$.

PAS 1 : Déterminer une solution optimale x^k de $\min_{x \in X} \{f(x) - \lambda_k g(x)\}$.

PAS 2 : Si $F(\lambda_k) = 0$, x^k est une solution optimale de (P) et λ_k est la valeur optimale. STOP.

PAS 3 : $\lambda_{k+1} = f(x^k)/g(x^k)$. Remplacer k par $k+1$ et retourner au pas 1.

CODE MATLAB

```
global lk;
f = 'fraction';
s = 'soustraction';
iter=0;

% PAS 0
x^0 =;
l(1) = feval(f,x^0);
k = 1;
lk = l(k);

while F <= -1 * 10^(-15)
        if k > 50
                break;
```

```
        end

% PAS 1
x^k = fminbnd('soustraction',a,b);
iter=iter+1;

% PAS 2
F = soustraction(x^k);
if F==0
        break;
end

% PAS 3
l(k+1)= fraction(x^k);
lk = l(k+1);
if l(k) - l(k+1) < 10^(-6)
        lk=l(k);
        break;
end
k = k+1;
end        % fin de la boucle while

disp('Le nombre d itération est');
iter
if F==0
        disp('La solution optimale est x = ');
         xk
        disp('La valeur optimale du prob est l = ');
        lk
else
        disp('la suite xk converge vers la valeur ');
        x^k
```

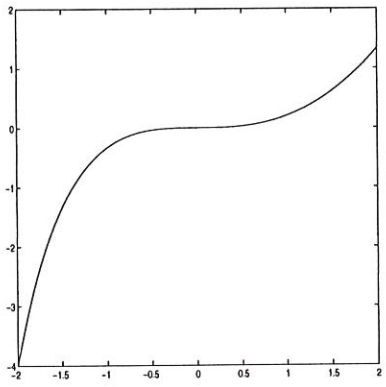
disp('la suite lk converge vers la valeur') ;

lk

end

Fin du programme

Exemple 24. Soit la fonction $f(x) = \frac{x^3}{x+4}$, $x \in [-2; 2]$ dessinée ci-dessous.



On utilise le programme avec les fonctions :

function f = fraction(x)

$f = \frac{x^3}{x+4}$;

function s = soustraction(x)

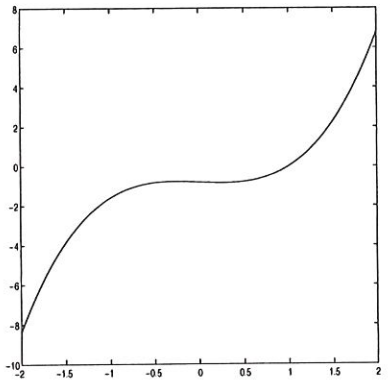
global lk ;

$s = x^3 - lk * (x + 4)$;

Prenons comme point de départ $x_0 = 1$ et comme bornes $a = -2$ et $b = 2$.

$$\lambda_1 = \frac{1^3}{1+4} = 0.2$$

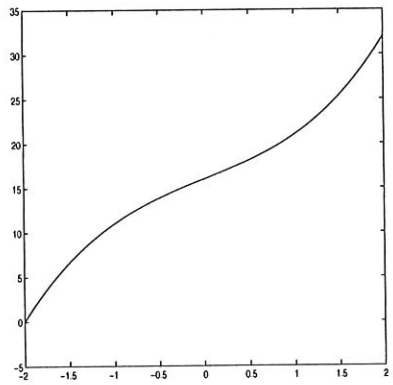
Minimisons la fonction $F(\lambda_1) = x^3 - 0.2 * (x + 4)$ dessinée ci-dessous.



La solution est $x_1 = -2$ et vaut $F = -8.4$. Puisque $F \neq 0$, on calcule λ_2 et on retourne au pas 1.

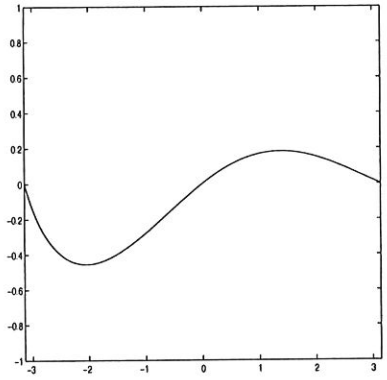
$$\lambda_2 = \frac{-2^3}{-2+4} = -4$$

On cherche x_2 le minimum de la fonction $F(\lambda_2) = x^3 + 4 * (x + 4)$ dessinée ci-dessous.



La solution est $x_2 = -2$ et vaut $F = 0$. Comme $F = 0$, l'algorithme s'arrête et nous rend comme solution $x = -2$ et comme valeur optimale -4, ce qui est juste.

Exemple 25. Soit la fonction $f(x) = \frac{\sin x}{x+4}$, $x \in [-\pi; \pi]$ dessinée ci-dessous.



Le minimum se trouve aux environs de $x = -2$ avec une valeur proche de -0.5

On utilise le programme avec les fonctions :

function f = fraction(x)

$f = \frac{\sin x}{x+4}$;

function s = soustraction(x)

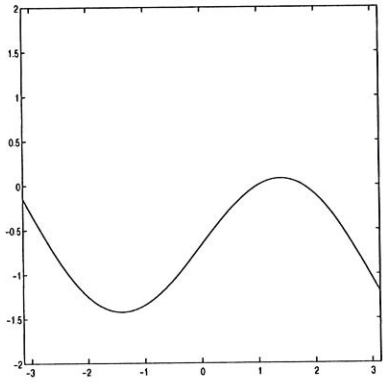
global lk ;

$s = \sin x - lk * (x + 4)$;

Prenons comme point de départ $x_0 = \frac{\pi}{2}$ et les bornes $a = -2$ et $b = 2$, et appliquons l'algorithme de Dinkelbach.

$$\lambda_1 = \frac{\sin x_0}{x_0 + 4} = 0.179508$$

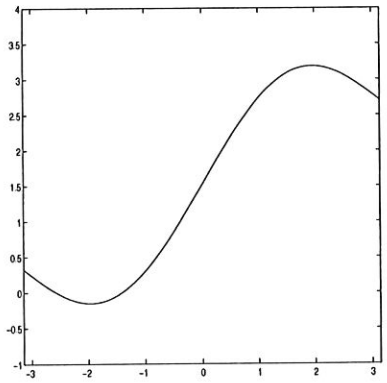
Minimisons la fonction $F(\lambda_1) = \sin x - \lambda_1 * (x + 4)$ dessinée ci-dessous.



La solution est $x_1 = -1.390310$ et vaut $F = -1.452216$. Puisque $F \neq 0$, on calcule λ_2 et on retourne au pas 1.

$$\lambda_2 = \frac{\sin x_1}{x_1 + 4} = -0.376963$$

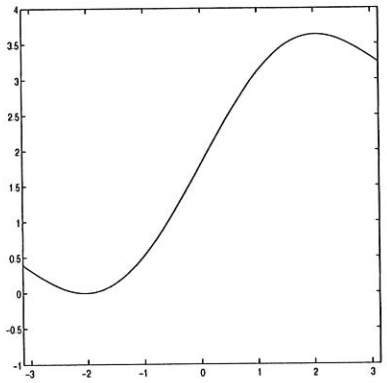
On cherche x_2 le minimum de la fonction $F(\lambda_2) = \sin x - \lambda_2 * (x + 4)$ dessinée ci-dessous.



La solution est $x_2 = -1.957310$ et vaut $F = -0.156210$. Puisque $F \neq 0$, on calcule λ_3 et on retourne au pas 1.

$$\lambda_3 = \frac{\sin x_2}{x_2 + 4} = -0.453436$$

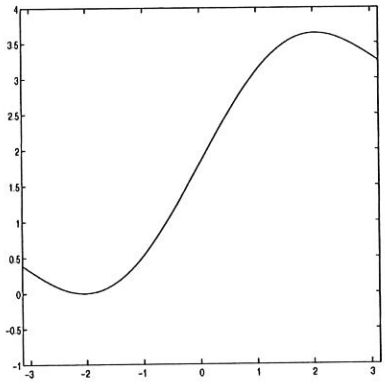
On cherche x_3 le minimum de la fonction $F(\lambda_3) = \sin x - \lambda_3 * (x + 4)$ dessinée ci-dessous.



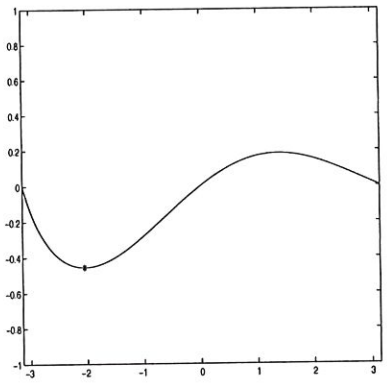
La solution est $x_3 = -2.041413$ et vaut $F = -0.003195$. Puisque $F \neq 0$, on calcule λ_4 et on retourne au pas 1.

$$\lambda_4 = \frac{\sin x_3}{x_3 + 4} = -0.455067$$

On cherche x_4 le minimum de la fonction $F(\lambda_4) = \sin x - \lambda_4 * (x + 4)$ dessinée ci-dessous.



La solution est $x_4 = -2.043244$ et vaut $F = -1.493920 * 10^{-6}$. On peut considérer que F est assez proche de 0 et s'arrêter là. Si on continuait, on aurait un $\lambda_5 = -0.455068$, c'est-à-dire qu'il ne décroit presque plus et donc que nous sommes déjà tout proche de la solution. Visualisons la solution trouvée sur le graphe.



3.2 Extension à plusieurs quotients.

L'algorithme de Dinkelbach pour T compact

PAS 0 : Soit $x^0 \in X$, $\lambda_1 = \max_{t\in T} \{f_t(x^0)/g_t(x^0)\}$, et $k = 1$.

PAS 1 : Déterminer une solution optimale x^k de

$$\inf_{x\in X} \left\{ \max_{t\in T} \{f_t(x) - \lambda_k g_t(x)\} \right\} \qquad (P_{\lambda_k}).$$

PAS 2 : Si $F(\lambda_k) = 0$, x^k est une solution optimale de (P) et λ_k est la valeur optimale. STOP.

PAS 3 : $\lambda_{k+1} = \max_{t\in T} \{f_t(x^k)/g_t(x^k)\}$. Remplacer k par $k+1$ et retourner au pas 1.

CODE MATLAB

```
global lk;
iter=0;
f='fraction';
s='soustraction';

%PAS 0
x^0 = ;
fraci = fraction(x^0);
l(1)= max(fraci);
lk=l(1);
k=1;

A = [ ];
b = [ ];
Aeq = [ ];
beq = [ ];
lb = [ ];
ub = [ ];
F=-1;

while F <= -1 * 10^(-15)
```

```
    if k>100
        break;
    end
```

%PAS 1

```
[xk,fval,maxfval] = fminimax('soustraction',x^0,A,b,Aeq,beq,lb,ub);
F = maxfval;
iter=iter+1;
```

%PAS 2

```
if F==0
    break;
end
```

%PAS 3

```
frac = fraction(xk);
l(k+1) = max(frac);
lk = l(k+1);

if l(k) - l(k+1) < 10^(-6)
    lk=l(k);
    break;
end

k = k+1;
end     % fin boucle while

disp('Le nombre d itération est');
iter
if F==0
    disp('La solution optimale x vaut');
    xk
```

```
        disp('et sa valeur optimale vaut') ;
        lk
else
        disp('x converge vers la valeur :') ;
        xk
        disp('lambda converge vers la valeur ;') ;
        lk
end
```

Fin du programme

Exemple 26.

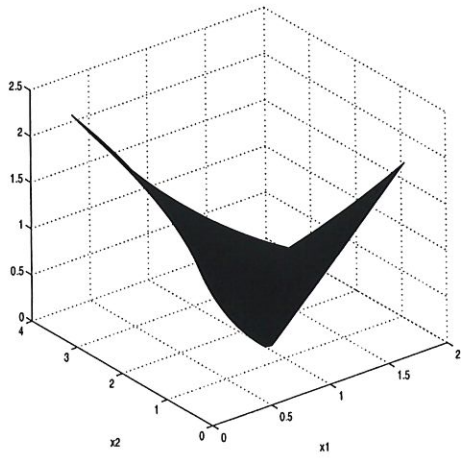
$$\min_{x \in X} \max \left\{ \frac{4x_1^3 + 11x_2}{16x_1 + 4x_2}, \frac{4x_1^2 - x_1}{3x_1 + x_2} \right\}$$

où

$$X = \left\{ x \in \mathbb{R}^2 | x_1 + x_2 \geq 1, 2x_1 + x_2 \leq 4, x_1, x_2 \geq 0 \right\}$$

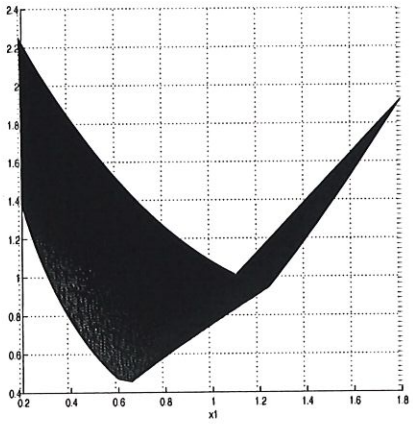
et le point initial est $x_0 = (1, 1)^T$.

Recherchons sur les graphiques la solution du problème 1. Dessinons la fonction $\max \left\{ \frac{4x_1^3+11x_2}{16x_1+4x_2}, \frac{4x_1^2-x_1}{3x_1+x_2} \right\}$ sur l'espace des contraintes X.



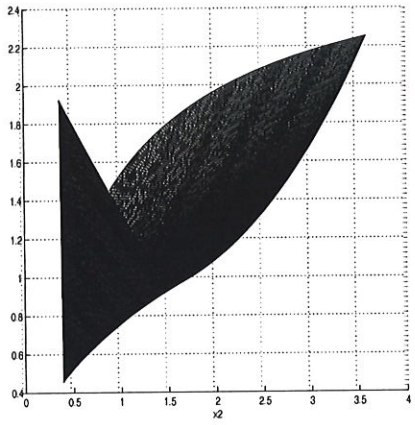
Essayons de trouver pour quelle valeur de x_1 et x_2 le minimum est atteint.

Voici la fonction vue du côté x_1.



On peut voir approximativement que le minimum est atteint pour une valeur de x_1 comprise entre 0.6 et 0.8 et vaut entre 0.4 et 0.6.

Voici la fonction vue du côté x_2.



On peut voir approximativement que le minimum est compris entre 0.4 et 0.5 pour une valeur de x_2 comprise entre 0.3 et 0.5.

Appliquons l'algorithme DTA avec $x^0 = [1;1]$, $A = [-1-1;21]$, $b = [-1;4]$ et $lb = [0;0]$ et les fonctions :

function f = fraction(x)

f(1) = (4*x(1)*x(1)*x(1)+11*x(2))/(16*x(1)+4*x(2)) ;

f(2) = (4*x(1)*x(1)-x(1))/(3*x(1)+x(2)) ;

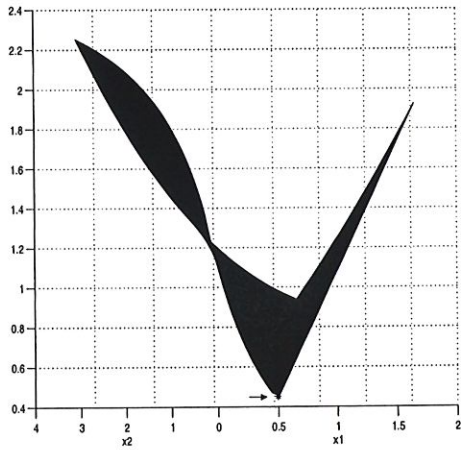
function s =soustraction(x)

global lk ;

s(1) = (4*x(1)*x(1)*x(1)+11*x(2))-lk*(16*x(1)+4*x(2)) ;

s(2) = (4*x(1)*x(1)-x(1))-lk*(3*x(1)+x(2)) ;

Nous obtenons la solution $\lambda = 0.432497$ au point $x = (0.636198; 0.363802)$ après 27 itérations. Visualisons la solution graphiquement.



Exemple 27.

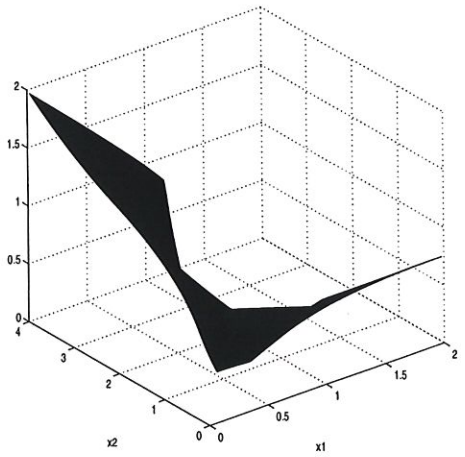
$$\min_{x \in X} \max \left\{ \left| \frac{3x_1 - 2x_2}{4x_1 + x_2} \right|, \left| \frac{x_1}{3x_1 + x_2} \right| \right\}$$

où

$$X = \left\{ x \in \mathbb{R}^2 | x_1 + x_2 \geq 1, 2x_1 + x_2 \leq 4, x_1, x_2 \geq 0 \right\}$$

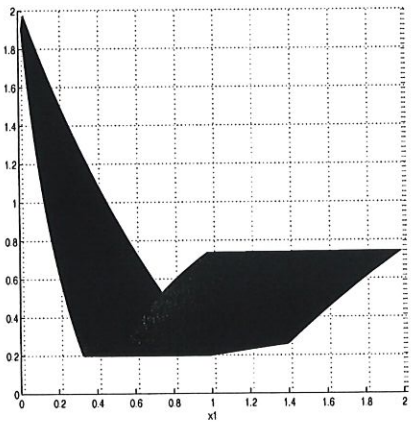
et le point initial est $x_0 = (1, 1)^T$.

Recherchons graphiquement la solution en dessinons la fonction $\max \left\{ \left| \frac{3x_1 - 2x_2}{4x_1 + x_2} \right|, \left| \frac{x_1}{3x_1 + x_2} \right| \right\}$ sur l'espace des contraintes X.

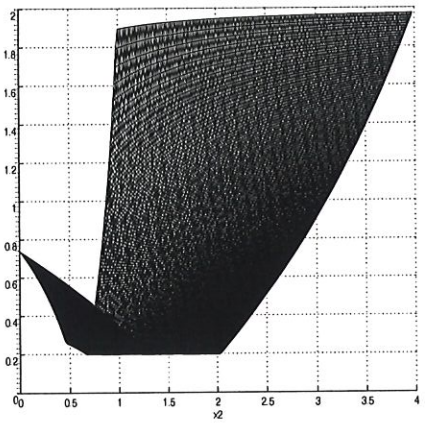


Essayons de trouver pour quelle valeur de x_1 et x_2 le minimum est atteint.

Voici la fonction vue du côté x_1.



On peut voir que le minimum vaut environ 0.2. pour une valeur de x_1 qu'on ne peut pas mieux borner que par [0.3,1].
Voici la fonction vue du côté x_2.

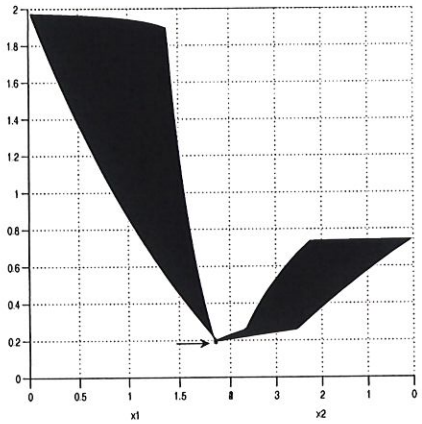


On peut voir approximativement que le minimum vaut environ 0.2 pour une valeur de x_2 qu'on ne peut pas mieux borner que par [0.7,2].

Appliquons l'algorithme DTA avec $x^0 = [1;1]$, $A = [-1 - 1;21]$, $b = [-1;4]$ et $lb = [0;0]$ et les fonctions :

```
function f = fraction(x)
f(1) = (3*x(1)-2*x(2))/(4*x(1)+x(2));
f(2) = -f(1);
f(3) = x(1)/(3*x(1)+x(2));
f(4) = -f(3);
function s = soustraction(x)
global lk;
s(1) = 3*x(1)-2*x(2)-lk*(4*x(1)+x(2));
s(2) = -3*x(1)+2*x(2)-lk*(4*x(1)+x(2));
s(3) = x(1)-lk*(3*x(1)+x(2));
s(4) = -x(1)-lk*(3*x(1)+x(2));
```

Nous obtenons la solution $\lambda = 0.196153$ au point $x = (0.573597; 1.203451)$ après 7 itérations. Visualisons la solution sur le graphique.



Exemple 28.

$$\min_{x \in X} \max_{i=0,\cdots,9} \left| \frac{8^4 x_1 + 8i^3 x_2 - i^4 x_3 - 8^3 i x_4}{8^4 x_4 + 8i^3 x_3} \right|$$

où

$$X = \left\{ x \in \mathbb{R}^4 | -1000 \leq x_1, x_2 \leq 1000, 1 \leq \frac{i^3 x_3 + 8^3 x_4}{8^3} \leq 1000, i = 0, \cdots, 8 \right\}$$

et le point initial est $x_0 = (0.5, 0, 0, 1)^T$.

Appliquons l'algorithme DTA avec $x^0 = (0.5, 0, 0, 1)$,

$$A = \left[0\ 0\ 0\ 8^3; 0\ 0\ 1\ 8^3; 0\ 0\ 2^3\ 8^3; 0\ 0\ 3^3\ 8^3; 0\ 0\ 4^3\ 8^3; 0\ 0\ 5^3\ 8^3; 0\ 0\ 6^3\ 8^3; 0\ 0\ 7^3\ 8^3; 0\ 0\ 8^3\ 8^3; 0\ 0\ 9^3\ 8^3;\right.$$

$$0\ 0\ 0 - 8^3; 0\ 0\ -1\ -8^3; 0\ 0\ -2^3 - 8^3; 0\ 0\ -3^3 - 8^3; 0\ 0\ -4^3 - 8^3; 0\ 0\ -5^3 - 8^3;$$

$$\left. 0\ 0\ -6^3 - 8^3; 0\ 0\ -7^3 - 8^3; 0\ 0\ -8^3 - 8^3; 0\ 0\ 9^3 - 8^3 \right],$$

$$b = \left[1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3; 1000 * 8^3\right.$$

$$\left. -8^3; -8^3; -8^3; -8^3; -8^3; -8^3; -8^3; -8^3; -8^3; -8^3 \right]$$

$lb = [-1000; -1000; -\infty; -\infty]$ et $up = [1000; 1000; +\infty; +\infty]$.

et les fonctions

function f = fraction(x)

$$f(1) = (8^4 * x(1) + 8 * 1^3 * x(2) - 1^4 * x(3) - 8^3 * 1 * x(4))/(8^4 * x(4) + 8 * 1^3 * x(3)) \quad f(10) = -f(1)$$
$$f(2) = (8^4 * x(1) + 8 * 2^3 * x(2) - 2^4 * x(3) - 8^3 * 2 * x(4))/(8^4 * x(4) + 8 * 2^3 * x(3)) \quad f(11) = -f(2)$$
$$f(3) = (8^4 * x(1) + 8 * 3^3 * x(2) - 3^4 * x(3) - 8^3 * 3 * x(4))/(8^4 * x(4) + 8 * 3^3 * x(3)) \quad f(12) = -f(3)$$
$$f(4) = (8^4 * x(1) + 8 * 4^3 * x(2) - 4^4 * x(3) - 8^3 * 4 * x(4))/(8^4 * x(4) + 8 * 4^3 * x(3)) \quad f(13) = -f(4)$$
$$f(5) = (8^4 * x(1) + 8 * 5^3 * x(2) - 5^4 * x(3) - 8^3 * 5 * x(4))/(8^4 * x(4) + 8 * 5^3 * x(3)) \quad f(14) = -f(5)$$
$$f(6) = (8^4 * x(1) + 8 * 6^3 * x(2) - 6^4 * x(3) - 8^3 * 6 * x(4))/(8^4 * x(4) + 8 * 6^3 * x(3)) \quad f(15) = -f(6)$$
$$f(7) = (8^4 * x(1) + 8 * 7^3 * x(2) - 7^4 * x(3) - 8^3 * 7 * x(4))/(8^4 * x(4) + 8 * 7^3 * x(3)) \quad f(16) = -f(7)$$
$$f(8) = (8^4 * x(1) + 8 * 8^3 * x(2) - 8^4 * x(3) - 8^3 * 8 * x(4))/(8^4 * x(4) + 8 * 8^3 * x(3)) \quad f(17) = -f(8)$$
$$f(9) = (8^4 * x(1) + 8 * 0^3 * x(2) - 0^4 * x(3) - 8^3 * 0 * x(4))/(8^4 * x(4) + 8 * 0^3 * x(3)) \quad f(18) = -f(9)$$

function s = soustraction(x)

global lk;

$$s(1) = (8^4 * x(1) + 8 * 1^3 * x(2) - 1^4 * x(3) - 8^3 * 1 * x(4)) - lk * (8^4 * x(4) + 8 * 1^3 * x(3));$$
$$s(2) = (8^4 * x(1) + 8 * 2^3 * x(2) - 2^4 * x(3) - 8^3 * 2 * x(4)) - lk * (8^4 * x(4) + 8 * 2^3 * x(3));$$
$$s(3) = (8^4 * x(1) + 8 * 3^3 * x(2) - 3^4 * x(3) - 8^3 * 3 * x(4)) - lk * (8^4 * x(4) + 8 * 3^3 * x(3));$$
$$s(4) = (8^4 * x(1) + 8 * 4^3 * x(2) - 4^4 * x(3) - 8^3 * 4 * x(4)) - lk * (8^4 * x(4) + 8 * 4^3 * x(3));$$
$$s(5) = (8^4 * x(1) + 8 * 5^3 * x(2) - 5^4 * x(3) - 8^3 * 5 * x(4)) - lk * (8^4 * x(4) + 8 * 5^3 * x(3));$$
$$s(6) = (8^4 * x(1) + 8 * 6^3 * x(2) - 6^4 * x(3) - 8^3 * 6 * x(4)) - lk * (8^4 * x(4) + 8 * 6^3 * x(3));$$
$$s(7) = (8^4 * x(1) + 8 * 7^3 * x(2) - 7^4 * x(3) - 8^3 * 7 * x(4)) - lk * (8^4 * x(4) + 8 * 7^3 * x(3));$$
$$s(8) = (8^4 * x(1) + 8 * 8^3 * x(2) - 8^4 * x(3) - 8^3 * 8 * x(4)) - lk * (8^4 * x(4) + 8 * 8^3 * x(3));$$
$$s(9) = (8^4 * x(1) + 8 * 0^3 * x(2) - 0^4 * x(3) - 8^3 * 0 * x(4)) - lk * (8^4 * x(4) + 8 * 0^3 * x(3));$$
$$s(10) = -(8^4 * x(1) + 8 * 1^3 * x(2) - 1^4 * x(3) - 8^3 * 1 * x(4)) - lk * (8^4 * x(4) + 8 * 1^3 * x(3));$$
$$s(11) = -(8^4 * x(1) + 8 * 2^3 * x(2) - 2^4 * x(3) - 8^3 * 2 * x(4)) - lk * (8^4 * x(4) + 8 * 2^3 * x(3));$$
$$s(12) = -(8^4 * x(1) + 8 * 3^3 * x(2) - 3^4 * x(3) - 8^3 * 3 * x(4)) - lk * (8^4 * x(4) + 8 * 3^3 * x(3));$$
$$s(13) = -(8^4 * x(1) + 8 * 4^3 * x(2) - 4^4 * x(3) - 8^3 * 4 * x(4)) - lk * (8^4 * x(4) + 8 * 4^3 * x(3));$$
$$s(14) = -(8^4 * x(1) + 8 * 5^3 * x(2) - 5^4 * x(3) - 8^3 * 5 * x(4)) - lk * (8^4 * x(4) + 8 * 5^3 * x(3));$$
$$s(15) = -(8^4 * x(1) + 8 * 6^3 * x(2) - 6^4 * x(3) - 8^3 * 6 * x(4)) - lk * (8^4 * x(4) + 8 * 6^3 * x(3));$$
$$s(16) = -(8^4 * x(1) + 8 * 7^3 * x(2) - 7^4 * x(3) - 8^3 * 7 * x(4)) - lk * (8^4 * x(4) + 8 * 7^3 * x(3));$$
$$s(17) = -(8^4 * x(1) + 8 * 8^3 * x(2) - 8^4 * x(3) - 8^3 * 8 * x(4)) - lk * (8^4 * x(4) + 8 * 8^3 * x(3));$$
$$s(18) = -(8^4 * x(1) + 8 * 0^3 * x(2) - 0^4 * x(3) - 8^3 * 0 * x(4)) - lk * (8^4 * x(4) + 8 * 0^3 * x(3));$$

Nous obtenons la solution $\lambda = 0.074185$ au point $x = (0.074178; 7.830533; 7.538106; 1)$ après 57 itérations.

3.3 Convergence superlinéaire.

L'algorithme modifié de Dinkelbach pour T compact

PAS 0 : Soit $x^0 \in X$, $\lambda_1 = \max_{t\in T} \{f_t(x^0)/g_t(x^0)\}$, et $k = 1$.

PAS 1 : Déterminer une solution optimale x^k de

$$F_k(\lambda_k) = \min_{x\in X} \left\{ \max_{t\in T} \left\{ \frac{f_t(x) - \lambda_k g_t(x)}{g_t(x^{k-1})} \right\} \right\} \qquad (Q_k).$$

PAS 2 : Si $F(\lambda_k) = 0$, x^k est une solution optimale de (P) et λ_k est la valeur optimale. STOP.

PAS 3 : $\lambda_{k+1} = \max_{t\in T} \{f_t(x^k)/g_t(x^k)\}$. Remplacer k par $k+1$ et retourner au pas 1.

CODE MATLAB

```
global lk;
global gk;
iter=0;
f='fraction';
s='msoustraction';

%PAS 0
x^0 = [];
fraci = fraction(x^0);
l(1)= max(fraci);
lk=l(1);
k=1;

A = [ ];
b = [ ];
Aeq = [ ];
```

beq = [ ];
lb = [ ];
ub = [ ];
F=-1;

while $F \leq -1 * 10^{(-15)}$
    if k>100
        break;
    end

%PAS 1

gk = gunction(x0)
[xk,fval,maxfval] = fminimax('msoustraction',x^0,A,b,Aeq,beq,lb,ub);
F = maxfval;
iter=iter+1;

%PAS 2

if F==0
    break;
end

%PAS 3

frac = fraction(xk);
l(k+1) = max(frac);
lk = l(k+1);

if $l(k) - l(k+1) < 10^{(-6)}$
    lk=l(k);
    break;
end

k = k+1;

```
end        % fin boucle while

disp('Le nombre d itération est') ;
iter
if F==0
        disp('La solution optimale x vaut') ;
        xk
        disp('et sa valeur optimale vaut') ;
        lk
else
        disp('x converge vers la valeur :') ;
        xk
        disp('lambda converge vers la valeur ;') ;
        lk
end
```

Fin du programme

Exemple 29. Reprenons l'exemple 4 et appliquons lui l'algorithme modifié. Les changements sont les fonctions :

```
function s = msoustraction(x)
global lk ;
global gk ;
s(1) = ((4*x(1)*x(1)*x(1)+11*x(2))-lk*(16*x(1)+4*x(2)))/gk(1) ;
s(2) = ((4*x(1)*x(1)-x(1))-lk*(3*x(1)+x(2)))/gk(2) ;
function g = gunction(x)
global lk ;
g(1) = 16*x(1)+4*x(2) ;
g(2) = 3*x(1)+x(2) ;
```

La solution obtenue est 0.432495 en x=(0.636200,0.363800) en 3 itérations seulement.

Exemple 30. Reprenons l'exemple 5 et appliquons lui l'algorithme modifié. Les changements sont les fonctions :

```
function s = msoustraction(x)
global lk;
global gk;
s(1) = (3*x(1)-2*x(2)-lk*(4*x(1)+x(2)))/gk(1);
s(2) = (-3*x(1)+2*x(2)-lk*(4*x(1)+x(2)))/gk(2);
s(3) = (x(1)-lk*(3*x(1)+x(2)))/gk(3);
s(4) = (-x(1)-lk*(3*x(1)+x(2)))/gk(4);
function g = gunction4(x)
global lk;
g(1) =4*x(1)+x(2);
g(2) = g(1);
g(3) = 3*x(1)+x(2);
g(4) = g(3);
```

La solution obtenue est 0.196152 en x=(0.576841,1.210256) en 3 itérations seulement.

Comparons les résultats obtenus avec l'algorithme de départ et l'algorithme modifié.

exemple	DTA	$MDTA$
exemple $4-7$	$\lambda = 0.432497$	0.432495
	$x_1 = 0.636198$	$x_1 = 0.636200$
	$x_2 = 0.363802$	$x_2 = 0.363800$
	iter $= 27$	iter $= 3$
exemple $5-8$	$\lambda = 0.196153$	$\lambda = 0.196152$
	$x_1 = 0.573597$	$x_1 = 0.576841$
	$x_2 = 1.203451$	$x_2 = 1.210256$
	iter $= 7$	iter $= 3$

3.4 Les algorithmes de type min-max semi-infini.

L'algorithme MDER

PAS 1 : Soient $x^0 \in X$, $t_1 \in T$, et $\epsilon, \eta, \gamma > 0$, $0 < \delta < 1$ donnés, calculer $\lambda_1 = \max_{t \in T} \{f_t(x^0)/g_t(x^0)\}$. Poser $m = 1$, $T_1 = \{t_1\}$, $k = 1$, $j = 1$, $x^m_{\epsilon,\lambda_k} = x^0$.

PAS 2 : Prendre $x^m_{\epsilon,\lambda_k} \in X$ qui satisfait

$$F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) \leq F^m_{\epsilon,\lambda_k}(x) + \delta^j, \qquad \forall x \in X,$$

et calculer $F_{\lambda_k}(x^m_{\epsilon,\lambda_k})$.

PAS 3 : Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) < -\eta$, mettre à jour le paramètre

$$\lambda_{k+1} = \max_{t \in T} \{f_t(x^m_{\epsilon,\lambda_k})/g_t(x^m_{\epsilon,\lambda_k})\};$$

poser $k = k + 1$, et retourner au pas 2.

PAS 4 : Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) \geq -\eta$, améliorer si nécessaire, m et ϵ pour une meilleure approximation.

PAS 4.1 : Si $F_{\lambda_k}(x^m_{\epsilon,\lambda_k}) > F^m_{\epsilon,\lambda_k}(x^m_{\epsilon,\lambda_k}) + \delta_k$, prendre

$$t_{m+1} \in \arg\max_{t \in T} \{f_t(x^m_{\epsilon,\lambda_k}) - \lambda_k g_t(x^m_{\epsilon,\lambda_k})\},$$

Diviser ϵ par un facteur constant, poser $T_{m+1} = T_m \cup \{t_{m+1}\}$, $j = j + 1$, et $m = m + 1$; retourner au pas 2.

PAS 4.2 : Si $\delta^j + \epsilon \log m > \gamma - \delta_k$, diviser ϵ par un facteur constant, poser $j = j + 1$ et retourner au pas 2.

PAS 5 : Si $\left|F_{\lambda_k}(x^m_{\epsilon,\lambda_k})\right| > \gamma$, calculer

$$\lambda_{k+1} = \max_{t \in T} \{f_t(x^m_{\epsilon,\lambda_k})/g_t(x^m_{\epsilon,\lambda_k})\},$$

poser $k = k + 1$, $j = 1$, et retourner au pas 2. Sinon, stopper et rendre x^m_{ϵ,λ_k} comme solution optimale à γ près du problème (P).

CODE MATLAB

%PAS 1

global xem ;

global T ;

global m ;

global lk ;

global e ;

x_0= ;

e=10 ;

$n = 10^{(-5)}$;

$g = 10^{(-5)}$;

d=0 ;

xem=x0 ;

[$t, fval$]=fminbnd(@fracMDER,0,1) ;

l(1)=-fval ;

lk=l(1)

m=1 ;

T(1)=t ;

k=1 ;

j=1 ;

iter=0 ;

%PAS 2

A=[] ;

B=[] ;

Aeq=[] ;

beq=[] ;

lb=[] ;

ub=[] ;

[xem,fem]=fmincon(@Fem,xem,A,B,Aeq,beq,lb,ub) ;

[t, $fval$]=fminbnd(@sousMDER,0,1) ;

rk=-fval

iter=iter+1 ;

while $((rk < -n)|(rk > fem + d^k)|(e * log(m) > g - d^k)|(abs(rk) > g))$

%PAS 3

if rk<-n

[t,fval]=fminbnd(@fracMDER,0,1) ;

l(k+1)=-fval ;

lk=l(k+1) ;

k=k+1

[xem,fem]=fmincon(@Fem,xem,A,B,Aeq,beq,lb,ub) ;

[t,fval]=fminbnd(@sousMDER,0,1) ;

rk=-fval

iter=iter+1 ;

%PAS 4.1

elseif $(rk > fem + d^k)$

tt=fminbnd(@sousMDER,0,1) ;

e=e/2 ;

T(m+1)=tt ;

j=j+1 ;

m=m+1 ;

[xem,fem]=fmincon(@Fem,xem,A,B,Aeq,beq,lb,ub) ;

[t,fval]=fminbnd(@sousMDER,0,1) ;

rk=-fval ;

iter=iter+1 ;

%PAS 4.2

elseif $(e * log(m) > g - d^k)$

e=e/2 ;

j=j+1 ;

[xem,fem]=fmincon(@Fem,xem,A,B,Aeq,beq,lb,ub) ;

[t,fval]=fminbnd(@sousMDER,0,1) ;

rk=-fval ;

iter=iter+1 ;

%PAS 5

elseif (abs(rk)>g)

[t,fval]=fminbnd(@fracMDER,0,1) ;

lk=-fval ;

k=k+1 ;

j=1 ;

[xem,fem]=fmincon(@Fem,xem,A,B,Aeq,beq,lb,ub) ;

[t,fval]=fminbnd(@sousMDER,0,1) ;

rk=-fval ;

iter=iter+1 ;

end

end

format long

sol=fracMder(t)

disp('L'agorithme converge vers la solution') ;

sol

disp('au point x=') ;

xem

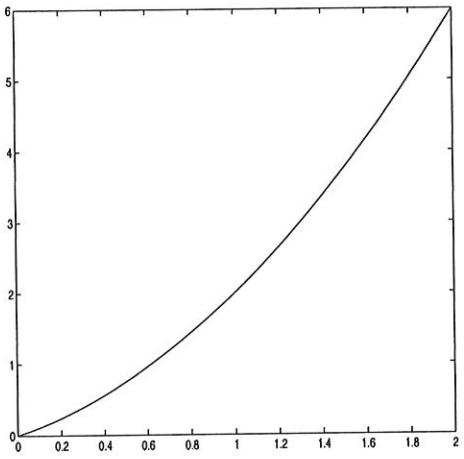
disp('Le nombre d'itérations est') ;

iter

Fin du programme

Exemple 31.

Cherchons la solution $\lambda = \min_{x\in X} \max_{t\in T} \frac{x^2+x}{t}$ où $X = [0,2]$ et $T = [1,2]$. On remarque de suite que la fonction est positive et qu'elle atteindra son maximum pour un t le plus petit possible c'est-à-dire pour $t = 1$. Le problème devient alors de minimiser la fonction $x^2 + x$ sur $[0,2]$ dessinée ci-dessous.



Il est évident que la solution est 0 et est obtenue pour $x = 0$. Regardons comment l'algorithme s'y prend pour résoudre le problème.

PAS 1 : Prenons comme point de départ $x_0 = 0.5$, $t_1 = 1.5$. Posons $m = 1$, $T_1 = \{t_1\}$, $k = 1$, $j = 1$ et $x^m_{\epsilon,\lambda_k} = x_0$.

$$\lambda_1 = \max_{t\in T} \frac{x_0^2 + x_0}{t} = \max_{t\in T} \frac{0.75}{t} = 0.75.$$

lb=[0.5 ;0.5] ; ub=[30 ;100] ;

PAS 2 : Cherchons

$$\begin{aligned} x_2 &= \min_{x\in X} F^m_{\epsilon,\lambda_1}(x) \\ &= \min_{x\in X} \epsilon \log\left(\sum_{t\in T_m} e^{(x^2+x-\lambda_1 t)/\epsilon}\right) \\ &= \min_{x\in X} \epsilon \log\left(e^{(x^2+x-\lambda_1 t_1)/\epsilon}\right) \\ &= \min_{x\in X} \epsilon(x^2+x-\lambda_1 t_1)/\epsilon \\ &= \min_{x\in X}(x^2+x-\lambda_1 * 1.5) \\ &= 0 \end{aligned}$$

$$F^m_{\epsilon,\lambda_1} x_1^m = \max_{t\in T}(x_1^m)^2 + x^m_{\epsilon,\lambda_1} - \lambda_1 t) = \max_{t\in T}(-0.75t) = -0.75$$

$$F^m_{\epsilon,\lambda_k}(x_2) = -0.75 < -10^{-5} = \eta \Rightarrow \text{PAS 3}$$

PAS 3 : $\lambda_2 = \max_{t\in T}\left\{(x_2^2+x_2)/t\right\} = \max_{t\in T}\frac{0}{t} = 0.$

Posons $k = k+1 = 2.$

PAS 2 : Cherchons

$$\begin{aligned} x_2 &= \min_{x\in X} F^m_{\epsilon,\lambda_2}(x) \\ &= \min_{x\in X} \epsilon \log\left(\sum_{t\in T_m} e^{(x^2+x)/\epsilon}\right) \\ &= \min_{x\in X} \epsilon \log\left(e^{(x^2+x)/\epsilon}\right) \\ &= \min_{x\in X} \epsilon(x^2+x)/\epsilon \\ &= \min_{x\in X}(x^2+x) \\ &= 0 \end{aligned}$$

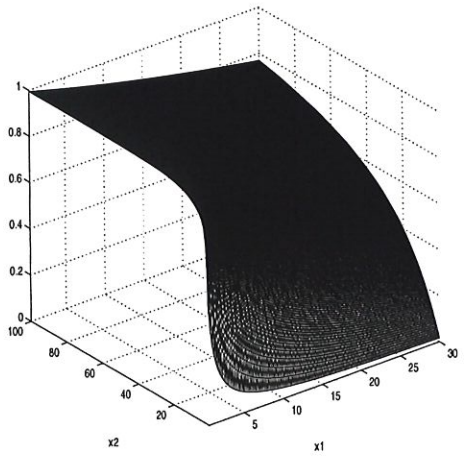
$$F^m_{\epsilon,\lambda_2} x_2^m = \max_{t\in T}(x_2^2 + x_2 - \lambda_2 t) = \max_{t\in T} 0 = 0.$$

$F_{\lambda_2}(x_2) = 0 \Rightarrow$ l'algorithme s'arrête. La solution optimale est $\lambda = 0$ en $x = 0.$

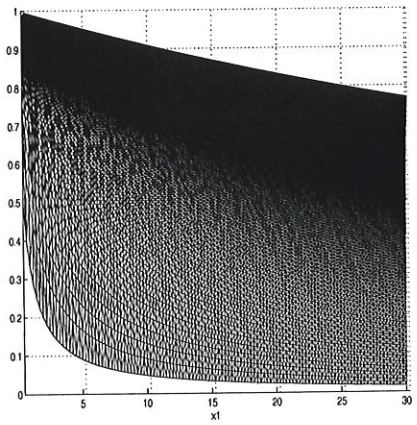
Exemple 32. Cherchons la solution de

$$\min_{x\in X}\max_{t\in T}\{(-tx_1+x_2)/(x_1+x_2)\},$$

où $X=[0.5,30]\times[0.5,100]$, et $T=[0,1]$. Il est évident que $\forall x\in X$, le maximum de la fonction est atteint lorsque t est la plus petite valeur de T, c'est-à-dire orsque $t=0$. Le problème est alors réduit à $\min_{x\in X}\{x_2/(x_1+x_2)\}$. Regardons sur le graphique où se situe le minimum.

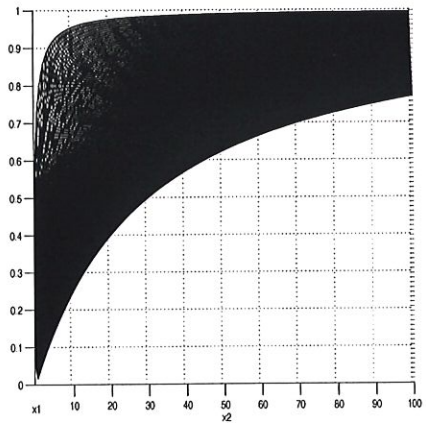


Voici une vue du graphique du côté x_1.



On peut remarquer que le minimum est atteint en $x_1=30$ et est plus petit que 0,05.

Voici une vue du graphique du côté x_2.



On peut remarquer que le minimum est atteint en $x_2$proche de zéro et est plus petit que 0,05.

Faisons tourner l'algorithme avec la valeur initiale $x_0 = (0.5, 100)$ et les fonctions :

function f= fracMDER(t)

global xem ;

$f = -((-t * xem(1) + xem(2))/(xem(1) + xem(2)))$;

function f=sousMDER(t)

global xem ;

global lk ;

$f = -(-t * xem(1) + xem(2) - lk * (xem(1) + xem(2)))$;

function f = Fem(x)

global T ;

global m ;

global lk ;

global e ;

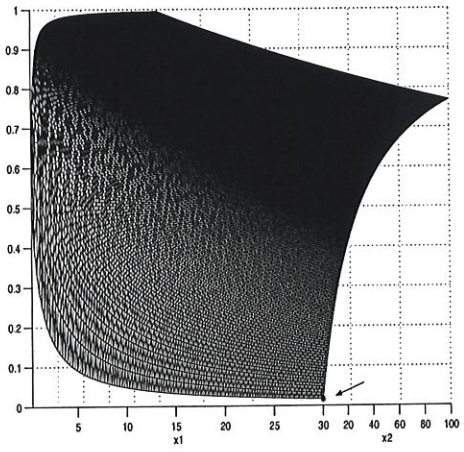
for i=1 :m

$\quad$ expo(i)=exp($(-T(i) * x(1) + x(2) - lk * (x(1) + x(2)))/e$) ;

end

f=e*log(sum(expo)) ;

La solution obtenue est $\lambda = 0.016393$ et $x = (30, 0.5)$ et est rendue après 2 itérations. Visualisons la solution sur le graphique.



Exemple 33. Le problème est défini par

$$\min_{x \in X} \max_{t \in T} \left\{ (t^2 x_1 x_2 + x_1^{2t} + t x_3^3) / \left[5(t-1)^2 x_1^4 + 2x_2^2 + 4t x_3 \right] \right\},$$

où $X = [0.5, 5] \times [0.5, 5] \times [0.5, 5]$, et $T = [0, 1]$.

Prenons comme point de départ $x_0 = (5, 5, 5)$, les autres paramètres étant choisis comme au problème 6 et les fonctions étant les suivantes :

function f= fracMDER(t)

global xem ;

$f = -(t^2 * xem(1) * xem(2) + xem(1)^{(2*t)} + t * xem(3)^3)/(5 * (t-1)^2 * xem(1)^4 + 2 * xem(2)^2 + 4 * t * xem(3))$;

function f=sousMDER(t)

global xem ;

global lk ;

$f = -((t^2 * xem(1) * xem(2) + xem(1)^{(2*t)} + t * xem(3)^3) - lk * (5 * (t-1)^2 * xem(1)^4 + 2 * xem(2)^2 + 4 * t * xem(3)))$;

function f = Fem(x)

global T ;

global m ;

global lk;

global e;

for i=1 :m

expo(i)=exp($(T(i)^2 * x(1) * x(2) + x(1)^{(2*T(i))} + T(i) * x(3)^3 - lk * (5 * (T(i) - 1)^2 * x(1)^4 + 2 *$

$x(2)^2 + 4 * T(i) * x(3)))/e$);

end

f=e*log(sum(expo));

L'agorithme nous donne après 11 itérations une solution $x = (0.5, 5, 0.5)$ qui vaut $\lambda = 0.055288$.

Conclusion

Dans ce mémoire nous avons étudié le problème suivant

$$(P) \qquad \bar{\lambda} = \min_{x \in X} \left\{ \max_{t \in T} \left\{ \frac{f_t(x)}{g_t(x)} \right\} \right\},$$

où X est un sous-ensemble compact non vide de $\mathbb{R}^n$, T est un espace métrique compact, $f_t(x), g_t(x)$ sont des fonctions continues sur $T \times \tilde{X}$ où $\tilde{X}$ est un sous-ensemble ouvert de $\mathbb{R}^n$ contenant X, et $g_t > 0$ sur X $\forall t \in T$.

Nous avons montré que résoudre ce problème revenait à chercher le zéro de la fonction paramétrique suivante

$$(P_\lambda) \qquad F(\lambda) = \min_{x \in X} \left\{ \max_{t \in T} \{ f_t(x) - \lambda g_t(x) \} \right\}.$$

Nous avons utilisé l'algorithme de Dinkelbach basé sur la méthode de Newton pour résoudre le problème dans le cas où T est un singleton et démontré que la convergence de cet algorithme était superlinéaire. Mais lorsque nous l'avons appliqué aux problèmes de type min-max à plusieurs quotients, la convergence obtenue était seulement linéaire. Nous avons alors opéré quelques changements afin que l'algorithme converge superlinéairement. Ensuite nous avons remarqué qu'il était parfois plus efficace de ne pas résoudre le sous-problème (P_λ) exactement. De plus, pour rendre l'algorithme implémentable dans le cas d'une infinité de quotients nous avons subdivisé T en sous-ensembles finis et utilisé la régularisation entropique qui approxime la fonction non différentiable $\max_{t \in T} \{ f_t(x) - \lambda g_t(x) \}$ par une fonction différentiable. Ainsi cette fonction différentiable peut facilement être minimisée par les méthodes quasi-Newton.

Bibliographie

[1] J.P. CROUZEIX, J.A. FERLAND et S.SCHAIBLE, *An Algorithm for Generalized Fractional Programs*, Journal of Optimization Theory and Applications, Vol. 47, pp. 35-49, 1985.

[2] W. DINKELBACH, *On Nonlinear Fractional Programming*, Management Science, Vol. 13, pp. 492-498, 1967.

[3] J.B. FRENK et S. SCHAIBLE, *Fractional Programming*, in *Handbook of Generalized Convexity and Generalized Monotonicity* (N. Hadjisavvas, S. Komlosi, S. Schaible), Chap. 8, pp. 335-386, 2005.

[4] J.Y. LIN et R.L. SHEU , *Solving Continuous Min-Max problems by an Iterative Entropic Regularization Method*, Journal of Optimization Theory and Applications, Vol. 121, No. 3, pp. 597-612, 2004.

[5] J.Y. LIN et R.L. SHEU , *Modified Dinkelbach-Type Algorithm for Generalized Fractional Programs with Infinitely Many Ratios*, Journal of Optimization Theory and Applications, Vol. 126, No. 2, pp. 323-343, 2005.

[6] J.J. STRODIOT, *Min-Max Fractional Programming*, CIUF-CUD Winter School on Optimization and Applied Mathematics, Hue University, Vietnam, December 11-21, 2005.